IBM DATA ENGINEERING PROFESSIONAL CERTIFICATE

Chương 1. Introduction to Data Engineering

# Kỹ thuật dữ liệu

Kỹ thuật dữ liệu bao gồm các nhiệm vụ:

* Thu thập dữ liệu nguồn bao gồm trích xuất, tích hợp và tổ chức dữ liệu từ các nguồn khác nhau. Thiết kế, xây dựng và duy trì kiến trúc dữ liệu có thể mở rộng để lưu trữ dữ liệu.
* Xử lý dữ liệu, sau khi thu thập được thông tin thì cần làm sạch dữ liệu (loại bỏ các dữ liệu thừa thải, không còn giá trị và chuyển đổi dữ liệu) để có thể sử dụng được dữ liệu đó.
* Bảo mật dữ liệu, quyền riêng tư.
* Lưu trữ dữ liệu sau khi làm sạch để dữ liệu luôn sẵn sàng được sử dụng. Đảm bảo các công cụ và hệ thống tuân theo quyền riêng tư, bảo mật, tuân thủ, giám sát, sao lưu và phục hồi dữ liệu.

Kỹ sư dữ liệu sử dụng kỹ thuật dữ liệu để có thể tạo ra một nguồn dữ liệu chất lượng cho những nhà khoa học dữ liệu, nhà phân tích kinh doanh, … có thể phân tích và phán đoán.

Để thành công trong vai trò của mình, kỹ sư dữ liệu cần có các kỹ năng kỹ thuật, chức năng cũng như kỹ năng mềm.

# Hệ sinh thái dữ liệu

Hệ sinh thái của kỹ sư dữ liệu bao gồm cơ sở hạ tầng, công cụ, khung và quy trình trích xuất dữ liệu, kiến trúc và quản lý đường ống dữ liệu và kho lưu trữ dữ liệu, quản lý quy trình làm việc, phát triển ứng dụng và quản lý BI và các công cụ Báo cáo.

Đầu tiên là dữ liệu, được chia thành: dữ liệu có cấu trúc (đạt tiêu chuẩn của những công cụ xử lý dữ liệu, có thể dùng luôn, ví dụ như file excel), dữ liệu bán cấu trúc, dữ liệu phi cấu trúc.

Tiếp theo là kho lưu trữ dữ liệu. Có hai loại kho dữ liệu chính: Giao dịch (chứa các thông tin giao dịch trực tuyến như ngân hàng) và Phân tích (các thông tin được sử dụng để phân tích).

Cuối cùng là các công cụ có thể truy vấn và thao tác với dữ liệu, ngôn ngữ truy vấn tiêu chuẩn hiện nay là SQL.

# Công nghệ kỹ thuật

- Spark streaming: xử lý dữ liệu

- RSS (Really simple syndication): nguồn cấp dữ liệu phổ biến

- Python

- Các thư viện hữu ích trong python: Numpy và scipy, pandas Beatifulsoup và Scrapy, Matplotlib và seaborn, Opency

- Ngôn ngữ R

- Java

- Unix/Linux

- Ngôn ngữ truy vấn: SQL

- ApacheHadoop, Apache Hive và Apache Spark: công cụ xử lý dữ liệu lớn (big data)

# Bảo mật dữ liệu

Ba thành phần chính để tạo ra một chiến lược hiệu quả cho bảo mật thông tin nói chung, thường được gọi là bộ ba CIA:

C (confidentiality): bảo mật bằng cách kiểm soát quyền truy cập

I (integrity): toàn vẹn bằng việc data có tính đáng tin cậy, không giả mạo

A (availability): sãn sàng bằng việc có thể truy cập ngay khi cần

Bốn khía cạnh hoặc mức độ bảo mật khác nhau:

Cấp độ đầu tiên là bảo mật cơ sở hạ tầng vật lý một thành phần quan trọng của bảo mật cho một hệ thống CNTT là bảo mật của vật lý; Cơ sở hạ tầng và cơ sở vật chất chứa Hệ thống trong trường hợp điện toán đám mây. Và bằng cách tính đến các mối đe dọa môi trường trước khi xem xét vị trí của cơ sở, ví dụ, các cơ sở hạ tầng không bao giờ được đặt ở vùng đồng bằng ngập lụt ở những khu vực dễ bị động đất, cơ sở hạ tầng được đặt trong cấu trúc chống động đất, chống sét đa cấp.

Cấp độ tiếp theo là an ninh mạng, an ninh mạng là rất quan trọng để giữ cho các hệ thống được kết nối với nhau và dữ liệu an toàn, các giải pháp bảo mật mạng bao gồm tường lửa để ngăn chặn truy cập trái phép vào các mạng riêng được kết nối với Internet Kiểm soát truy cập mạng để đảm bảo an ninh điểm cuối bằng cách chỉ cho phép các thiết bị được ủy quyền kết nối với mạng, ví dụ: mạng công ty có thể không cho phép các thiết bị có gói dịch vụ lỗi thời kết nối với phân đoạn mạng của chúng để tạo silo hoặc mạng cục bộ ảo trong mạng để bạn có thể tách biệt tài sản của mình thành các silo riêng lẻ dựa trên mức độ bảo mật cần thiết cho các giao thức bảo mật tài sản khác nhau để đảm bảo kẻ tấn công không thể khai thác dữ liệu trong khi nó đang được vận chuyển và phát hiện xâm nhập và hệ thống ngăn chặn xâm nhập Kiểm tra lưu lượng truy cập đến để tìm các nỗ lực và lỗ hổng xâm nhập

Khía cạnh thứ ba của bảo mật là bảo mật ứng dụng, Bảo mật ứng dụng là rất quan trọng để giữ dữ liệu khách hàng riêng tư và đảm bảo các ứng dụng nhanh và đáp ứng Bảo mật cần được tích hợp vào Nền tảng của một ứng dụng. Để ngăn chặn các ứng dụng và dịch vụ khác đưa ra các lỗ hổng, bạn có thể làm cho ứng dụng của mình an toàn bằng cách tuân theo các thực tiễn kỹ thuật bảo mật như mô hình hóa mối đe dọa để xác định các điểm yếu tương đối và các mẫu tấn công liên quan đến thiết kế bảo mật ứng dụng nhằm giảm thiểu rủi ro, hướng dẫn và thực hành mã hóa bảo mật ngăn chặn lỗ hổng và kiểm tra bảo mật để khắc phục sự cố trước khi ứng dụng được triển khai và để xác thực rằng ứng dụng không có vấn đề bảo mật đã biết

Và khía cạnh thứ tư và cuối cùng là bảo mật dữ liệu: dữ liệu được lưu trữ hoặc chuyển tiếp giữa các hệ thống, ứng dụng, dịch vụ và khối lượng công việc, có thể là ở trạng thái nghỉ hoặc chuyển động; dữ liệu cần được bảo vệ; một trong những điều khiển chính cho dữ liệu.

# Thu thập dữ liệu

Có một số phương pháp và công cụ có sẵn để thu thập dữ liệu. Chúng bao gồm các ngôn ngữ truy vấn để trích xuất dữ liệu từ cơ sở dữ liệu, API, Web Scraping, Data Streams, RSS Feeds và Data Exchanges.

Sau khi đã thu thập được dữ liệu thì cần làm sạch và sắp xếp dữ liệu để có thể sẵn sàng thực hiện sử dụng.

CHƯƠNG 2. Python for Data Science, AI & Development

# 2.1 Các kiểu dữ liệu và biểu thức

Các kiểu dữ liệu:  
 int là số nguyên, float là số thực, string là chuỗi, boolean là kiểu dữ liệu đúng sai, list là kiểu dữ liệu danh sách, dictionary là kiểu dữ liệu có ket và value, set là kiểu dữ liệu tự động loại bỏ các giá trị trùng nhau.

Các biểu thức có thể sử dụng trong Python:

+,-,\*,/,// và các toán tử khác như so sánh, gán, …

# 2.2 Cấu trúc dữ liệu Python

Ta có hàm range(x) trả về môt list có giá trị từ 0 đến x-1.

Các vòng lặp trong python là: for và while, trong đó vòng lặp for biết trước số lần lặp, còn vòng lặp while chưa biết số lần lặp cụ thể.

Câu lệnh rẽ nhánh if/if … elif

Câu lệnh bắt ngoại lệ try … except: nó sẽ in ra thông báo lỗi nếu xuất hiện lỗi trong lệnh try

Tạo hàm trong Python bằng câu lệnh def ten\_ham:

# 2.3 Làm việc với file trong Python

|  |
| --- |
| File1=open(“url\_file”, “mode”) |

Trong đó url\_file là đường dẫn đến file, mode là chế độ (đọc file, viết file, tạo file).

Mode là các chức năng thao tác với file, “r” là read, “w” là viết đè lên file, “a” là them vào file, không ghi đè lên các data cũ như “w”, “x” là tạo 1 file mới, và một số mode khác. Luôn đóng file khi đã sử dụng xong file, dung câu lệnh sau để đóng file: file1.close

Để tránh dính lỗi quên đóng file sau khi sử dụng thì khuyến khích dùng câu lệnh:

|  |
| --- |
| with open(“url\_file”, “mode”) as File1 |

Cách đọc file nội dung tiếng Việt:

|  |
| --- |
| File= open(“url\_file”,mode=”r”, encoding=”utf-8”) |

Cách lấy nội dung của file:

|  |
| --- |
| content = file.read()  print(content) |

Cách viết thêm dòng vào file:

|  |
| --- |
| with open('Example2.txt', 'w') as File1:  File1.write("This is line A\n")  File1.write("This is line B\n") |

Cách để sao chép 1 file vào 1 file khác:

|  |
| --- |
| # Open the source file for reading  with open('source.txt', 'r') as source\_file:  # Open the destination file for writing  with open('destination.txt', 'w') as destination\_file:  # Read lines from the source file and copy them to the destination file  for line in source\_file:  destination\_file.write(line)  # Destination file is automatically closed when the 'with' block exits  # Source file is automatically closed when the 'with' block exits |

# 2.4 Các thư viện phổ biến trong Python

- Beautifulsoup: Thư viện BeautifulSoup là một thư viện của Python cho phép chúng ta lấy dữ liệu từ HTML đơn giản và hiệu quả.

- Request: Requests module là một thư viện hỗ trợ chúng ta có thể gửi bất kỳ một loại request HTTP nào một cách đơn giản nhất.

- Pandas: Là thư viện dùng để thao tác với nhiều loại file khác nhau như csv, xls, dữ liệu ma trận, dữ liệu thời gian cố định, không cố định.

- Numpy: Là một thư viện toán học phổ biến và mạnh mẽ của Python. Cho phép làm việc hiệu quả với ma trận và mảng, đặc biệt là dữ liệu ma trận và mảng lớn với tốc độ xử lý nhanh hơn nhiều lần.

# 2.5 OOP trong Python

Cấu trúc của class:

|  |
| --- |
| class ClassName:  # tập lệnh |

Ví dụ:

|  |
| --- |
| class Employee:  id = 10  name = "The Mac"  def display (self):  print(self.id, self.name) |

self ở đây luôn là đối số đầu tiên trong hàm, dù nó có được sử dụng hay không, self sẽ trỏ đến đối tượng hiện tại

Một số hàm có sẵn cho class:

getattr(obj,name): trả về name của đối tượng obj

setattr(obj, name,value): gán value cho thuộc tính name của đối tượng obj

delattr(obj,name): xóa thuộc tính name của đối tượng obj

hasattr(obj, name): trả về true/false nếu thuộc tính name xuất hiện trong đối tượng obj

cách để khai báo public, protected, private cho các thuộc tính và hàm:

với public thì có thể truy cập được ở mọi nơi, nên cách khai báo như hàm bình thường

Với protected thì chỉ lớp con có thể truy cập được, cách khai báo bằng cách bắt đầu bằng một dấu gạch ngang "\_", VD: \_age.

Với private thì chỉ class đó có quyền truy cập, cách khai báo bằng cách bắt đầu bằng 2 dấu gạch ngang "\_\_", VD: \_\_age.

# 2.6 Làm việc với Pandas và webscraw

Đầu tiên cần imprort thư viện cần thiết:

|  |
| --- |
| import pandas as pd |

Để đọc file định dạng csv ta dùng lệnh pd.read\_csv

|  |
| --- |
| df = pd.read\_csv("addresses.csv", header=None) |

Từ đó có thể thao tác với tệp có định dạng csv bằng biến “df”.

Ví dụ:

|  |
| --- |
| df.loc[[0,1,2], "First Name" ] |

Câu lệnh này in dòng 1, 2, 3 của cột có tên “First Name”

Với các file excel ta dùng lệnh “pd.read\_excel”, file xml dùng “pd.read\_xml”, …

Để lưu file với định dạng khác ta dùng lệnh “to\_csv”, “to\_excel”, …

Ví dụ:

|  |
| --- |
| df.to\_csv("employee.csv", index=False) |

Lưu 1 dataframe có tên là employee.csv

Ta có bảng tóm tắt như sau:

|  |  |  |
| --- | --- | --- |
| Data formate | Read | Save |
| Csv | Pd.read\_csv() | Df.to\_csv() |
| Json | Pd.read\_json() | Df.to\_json() |
| Excel | Pd.read\_excel() | Df.to\_excel() |
| Hdf | Pd.read\_hdf() | Df.to\_hdf() |
| Sql | Pd.read\_sql() | Df.to\_sql() |
| … | … | … |

Để lấy dữ liệu từ trang web sử dụng thư viện sau:

|  |
| --- |
| from bs4 import BeautifulSoup  soup = BeautifulSoup(html, 'html5lib') |

Trong đó html là đường dẫn của web, ‘html5lib’ là định dạng mặc định

Để in ra các mã html của web sử dụng lệnh “soup.prettify()”, ví dụ:

|  |
| --- |
| print(soup.prettify()) |

Và các dòng lệnh title, h3, b để in ra các thẻ tương ứng. Cũng có thể sử dụng các lệnh “find”, type, get, attrs để lấy các thuộc tính của thẻ trang web.

CHƯƠNG 3. Python Project for Data Engineering

# 3.1 Các thư viện cần thiết

Để lấy dữ liệu từ các file csv, xml, json bao gồm các bảng có thông tin như sau:

|  |  |  |
| --- | --- | --- |
| Name | Height (inch) | Weight (pound) |
| … | … | … |

Đầu tiên cần import các thư viện trong Python

|  |
| --- |
| import glob  import pandas as pd  import xml.etree.ElementTree as ET  from datetime import datetime |

Thư viện glob để chọn file, thư viện pd, ET để xử lý file xml, … thư viện datetime để lấy ngày giờ.

|  |
| --- |
| tmpfile = "temp.tmp"  logfile = "logfile.txt"  targetfile = "transformed\_data.csv" |

Tạo 1 file để chứa dữ liệu extract, 1 file chứa nhật ký thya đổi của từng lệnh, 1 file chứa data

# 3.2 Extract

Hàm extract file csv, json, xml:

|  |
| --- |
| def extract\_from\_csv(file\_to\_process):  dataframe = pd.read\_csv(file\_to\_process)  return dataframe |

|  |
| --- |
| def extract\_from\_json(file\_to\_process):  dataframe = pd.read\_json(file\_to\_process,lines=True)  return dataframe |

|  |
| --- |
| def extract\_from\_xml(file\_to\_process):  dataframe = pd.DataFrame(columns=["name", "height", "weight"])  tree = ET.parse(file\_to\_process)  root = tree.getroot()  for person in root:  name = person.find("name").text  height = float(person.find("height").text)  weight = float(person.find("weight").text)  dataframe = dataframe.append({"name":name, "height":height, "weight":weight}, ignore\_index=True)  return dataframe |

Tạo hàm tự động giải nén file tùy vào định dạng:

|  |
| --- |
| def extract():  extracted\_data = pd.DataFrame(columns=['name','height','weight'])    #process all csv files  for csvfile in glob.glob("\*.csv"):  extracted\_data = extracted\_data.append(extract\_from\_csv(csvfile), ignore\_index=True)    #process all json files  for jsonfile in glob.glob("\*.json"):  extracted\_data = extracted\_data.append(extract\_from\_json(jsonfile), ignore\_index=True)    #process all xml files  for xmlfile in glob.glob("\*.xml"):  extracted\_data = extracted\_data.append(extract\_from\_xml(xmlfile), ignore\_index=True)    return extracted\_data |

# 3.3 Transform

|  |
| --- |
| def transform(data):  data.height = data.height.astype(float)  #(one inch is 0.0254 meters)  data['height'] = round(data.height \* 0.0254,2)    data.weight = data.weight.astype(float)  #(one pound is 0.45359237 kilograms)  data['weight'] = round(data.weight \* 0.45359237,2)  return data |

Chuyển đổi định dạng dữu liệu cho phù hợp, bao gồm:

Chuyển đổi kiểu dữ liệu thành float, và đổi inch sang mét, pound sang kilogram

# 3.4 Load

|  |
| --- |
| def load(targetfile,data\_to\_load):  data\_to\_load.to\_csv(targetfile) |

# 3.5 Log

Ghi lại nhật ký cho quá trình

|  |
| --- |
| def log(message):  timestamp\_format = '%Y-%h-%d-%H:%M:%S'  now = datetime.now()  timestamp = now.strftime(timestamp\_format)  with open("logfile.txt","a") as f:  f.write(timestamp + ',' + message + '\n') |

Để tối ưu hóa cũng như nhật ký rõ ràng hơn, nên đặt hàm log vào mỗi quá trình ETL với mỗi message tương ứng.

Ví dụ:

|  |
| --- |
| def load(targetfile,data\_to\_load):  log( “start load file”)  data\_to\_load.to\_csv(targetfile) |

Vậy là hoàn thành quy trình ETL, chạy thử nghiệm với các file csv, xml, json có trong thư mục:

|  |
| --- |
| log("Extract phase Started")  extracted\_data = extract()  log("Extract phase Ended")  print(extracted\_data) |

Đầu ra sẽ là 1 bảng chứ các cột name, height, weight với kiểu dữ liệu inch, pound.

|  |
| --- |
| log("Transform phase Started")  transformed\_data = transform(extracted\_data)  log("Transform phase Ended")  transformed\_data |

Chuyển đổi các kiểu dữ liệu về mét và kilogram

|  |
| --- |
| log("Load phase Started")  load(targetfile,transformed\_data)  log("Load phase Ended") |

Và lưu dữ liệu vào một file mới.

CHƯƠNG 4. Introduction to Relational Databases (RDBMS)

# 4.1 Tổng quan

Dữ liệu là gì? Là có thể bao gồm Sự kiện, quan sát, nhận thức, số, ký tự, ký hiệu, hình ảnh Hoặc kết hợp bất kỳ trong số này. Dữ liệu có thể được phân loại theo mức độ và độ cứng của cấu trúc của nó. Dữ liệu có thể được cấu trúc, bán cấu trúc hoặc không có cấu trúc. Dữ liệu có cấu trúc có thể được biểu diễn theo hàng và cột, giống như bảng. Nó có một lược đồ được xác định rõ ràng và cấu trúc. Dữ liệu bán cấu trúc có một số thuộc tính tổ chức, nhưng không đủ để dễ dàng lưu trữ trong các hàng và cột theo yêu cầu của lược đồ dạng bảng cứng. Thay vào đó, dữ liệu bán cấu trúc được tổ chức thành một hệ thống phân cấp bằng cách sử dụng thẻ và siêu dữ liệu. Dữ liệu phi cấu trúc không có cấu trúc có thể nhận dạng; nó không tuân theo bất kỳ định dạng, trình tự, ngữ nghĩa hoặc quy tắc cụ thể nào. Nó không thể được tổ chức thành định dạng bảng để lưu trữ trong cơ sở dữ liệu quan hệ. Dữ liệu phi cấu trúc thường được lưu trữ trong cơ sở dữ liệu NoSQL.

Sự khác nhau giữa information và data:

Information là đại diện trừu tượng gồm nhiều thuộc tính, mối quan hệ là thường thể hiện bằng sơ đồ phân cấp sử dụng cấu trúc cây. Còn data được xác định ở mức cụ thể hơn, gồm nhiều chi hiết và thường được thể hiện ở dạng bảng bao gồm hàng và cột

Hoặc có thể hiểu một cách đơn giản thì information là tập hợp các loại dữ liệu, còn data là tập hợp những information hữu ích với mỗi nhiệm vụ, nhu cầu cụ thể.

# 4.2 Mô hình quan hệ

Bao gồm những bảng có các thuộc tính trở thành cột trong một bảng. Các thuộc tính sẽ có các kiểu dữ liệu khác nhau như chữ, số, ngày giờ, giá trị đúng sai, …

Các mối quan giữa giữa các bảng có thể là một -một, một -nhiều, nhiều -nhiều

# 4.3 Các câu lệnh SQL

Ngôn ngữ SQL được chia làm hai, gồm ngôn ngữ tương tác với thực thể (SQL data definition language) và ngôn ngữ tương tác với thuộc tính (SQL data manipulation language)

Các câu lệnh phổ biến trong SQL DDL là:

CREATE: tạo bảng

ALTER: thay đổi bảng

TRUNCATE: xóa dữ liệu trong bảng không phải xóa chính bảnh

DROP: xóa bảng

Các câu lệnh phổ biến trong SQL DML:

INSERT: chèn 1 hoặc nhiều hàng vào bảng

SELECT: đọc hoặc chọn hàng từ bảng

UPDATE: chỉnh sửa hàng trong bảng

DELETE: xóa 2 hàng hoặc nhiều hàng dữ liệu ra khỏi bảng

Một số ràng buộc:

Khóa chính, khóa phụ, ràng buộc NULL, quan hệ giữa các bảng, …

# 4.4 Giới thiệu các cơ sở dữ liệu

## a. MySQL

Công cụ quản lý MySQL bằng phần mềm phpmyAdmin, MySQL Workbench là một công cụ thiết kế cơ sở dữ liệu trực quan tích hợp phát triển, quản trị, thiết kế cơ sở dữ liệu, tạo và bảo trì SQL vào một môi trường phát triển duy nhất cho hệ thống cơ sở dữ liệu MySQL.

Thao tác với phpmyadmin có giao diện thân thiện với người dùng, có thể cài đặt ON DELETE hoặc ON UPDATE khi cài khóa ngoại như một ràng buộc.

Để có thể backup hay restore dữu liệu MySQL thì sử dụng mysqldump. Mysqldum hỗ trợ backup khá tốt

## b. PosgreSQL

Có thể sử dụng pgAdmin để quản lý PostgreSQL ở dạng web hoặc phần mềm máy tính. Sử dụng pgAdmin, có thể:

Sử dụng pg\_dump để sao lưu cơ sở dữ liệu và psql để khôi phục. Sử dụng công cụ Nhập/Xuất để tải dữ liệu vào và xuất dữ liệu từ bảng.

Sử dụng chế độ xem:

Sử dụng chế độ xem để giới hạn quyền truy cập vào dữ liệu nhạy cảm và đơn giản hóa việc truy xuất dữ liệu. Chế độ xem có thể được cụ thể hóa, nghĩa là chế độ xem lưu trữ bộ kết quả được đặt để truy cập tiếp theo nhanh hơn.

Chế độ xem cụ thể hóa nâng cao hiệu suất vì chế độ xem được lưu và thường được lưu trữ trong bộ nhớ. Tuy nhiên, không thể chèn, cập nhật hoặc xóa hàng trong dạng xem cụ thể hóa và chúng phải được làm mới trước khi có thể xem dữ liệu cập nhật.

CHƯƠNG 5. Databases and SQL for Data Science with Python

## 5.1 Các câu lệnh thao tác với dữ liệu

Bảng thống kê:

|  |  |  |
| --- | --- | --- |
| Lệnh | Cấu trúc | Mô tả |
| SELECT | SELECT column1, column2,...  FROM table\_name; | Lấy dữ liệu ở các cột từ 1 bảng |
| WHERE | SELECT column1, column2,...  FROM table\_name  WHERE condition; | Điều kiện WHERE giúp dữ liệu lấy ra chính xác, cụ thể hơn |
| COUNT | SELECT COUNT \*  FROM table\_name ; | COUNT đếm giá trị các hàng trong bảng |
| DISTINCT | SELECT DISTINCT columnname  FROM table\_name; | DISTINCT giới hạn các giá trị trùng lặp trong kết quả lấy ra |
| LIMIT | SELECT \*  FROM table\_name  LIMIT number; | LIMIT giới hạn các giá trị được truy vấn trong bảng |
| TOP | SELECT TOP number columnname  FROM table\_name; | TOP giới hạn giá trị trả về khi truy vấn, khác với LIMIT |
| INSERT | INSERT INTO table\_name (column1,column2,column3...) VALUES(value1,value2,value3...); | INSERT chèn thêm 1 hàng có các giá trị vào bảng |
| UPDATE | UPDATE table\_name SET[[column1]=[VALUES]]  WHERE [condition]; | UPDATE cập nhật các giá trị với điều kiện đi kèm để chính xác nhất |
| DELETE | DELETE FROM table\_name  WHERE [condition]; | DELETE xóa một hàng từ bảng với điều kiện đi kèm |

Ngoài ra còn có thêm một số cú pháp như: SUM dúng để tính tổng đối với các số học, LENGTH, UPCASE, LCASE đối với chuổi, DAY/TIME, ORDER BY để sắp xếp dữ liệu sau khi truy vấn, GROUP BY để nhóm các dữ liệu cùng nhóm và mệnh đề HAVING đi kèm với GROUP BY

## 5.2 Các câu lệnh thao tác với thực thể

|  |  |  |
| --- | --- | --- |
| Lệnh | Cấu trúc | Mô tả |
| CREATE | CREATE TABLE table\_name (  COLUMN1 datatype,  COLUMN2 datatype,  COLUMN3 datatype,  …  ); | Tạo bảng CSDL |
| DROP | DROP TABLE table\_name | Xóa bảng |
| ALTER TABLE | ALTER TABLE table\_name  ADD COLUMN column\_name data\_type column\_constraint; | Thêm một cột vào bảng |
| ALTER TABLE | ALTER TABLE table\_name CHANGE column\_name  column\_name VARCHAR(20); | Thay đổi kiểu dữ liệu của một cột |
| TRUNCATE TABLE | TRUNCATE TABLE table\_name; | Xóa tất cả hàng trong bảng |

# 5.3 Câu lệnh lồng nhau

Câu lệnh lồng nhau là một truy vấn bên trong một truy vấn SQL khác và được nhúng trong mệnh đề WHERE. Truy vấn con được sử dụng để trả về dữ liệu sẽ được sử dụng trong truy vấn chính làm điều kiện để hạn chế hơn nữa dữ liệu được truy xuất

Ví dụ:

|  |
| --- |
| SELECT column\_name [, column\_name ]  FROM table1 [, table2 ]  WHERE column\_name OPERATOR (SELECT column\_name [, column\_name ] FROM table1 [, table2 ] [WHERE]) |

|  |
| --- |
| SELECT emp\_id, fmame, lname, salary  FROM employees  where salary < (SELECT AVG(salary) FROM employees); |

Để truy vấn kết hợp hai hoặc nhiều bản ghi nhưng chỉ hiển thị các giá trị trùng khớp trong cả hai bảng. Kết nối bên trong chỉ áp dụng cho các cột được chỉ định.

|  |
| --- |
| SELECT column\_name(s)  FROM table1, table2  WHERE table1.column\_name = table2.column\_name; |

Ví dụ như:

|  |
| --- |
| SELECT \*  FROM employees, jobs  WHERE employees.job\_id = jobs.job\_ident; |

# 5.4 Truy cập CSDL bằng Python

Sử dụng với sqlite3

Đầu tiên import thư viện và kết nối với CSDL:

|  |
| --- |
| import sqlite3  conn = sqlite3.connect('INSTRUCTOR.db') |

Thao tác với với CSDL thông qua con trỏ được khai báo:

|  |
| --- |
| cursor\_obj = conn.cursor() |

Trước khi tạo bảng cần xóa bảng, nếu đã tồn tại.

|  |
| --- |
| cursor\_obj.execute("DROP TABLE IF EXISTS INSTRUCTOR")  table = """ create table IF NOT EXISTS INSTRUCTOR(ID INTEGER PRIMARY KEY NOT NULL, FNAME VARCHAR(20), LNAME VARCHAR(20), CITY VARCHAR(20), CCODE CHAR(2));"""    cursor\_obj.execute(table)    print("Table is Ready") |

Sau khi tạo bảng, thêm giá trị các hàng vào bảng:

|  |
| --- |
| cursor\_obj.execute('''insert into INSTRUCTOR values (1, 'Rav', 'Ahuja', 'TORONTO', 'CA')''')  cursor\_obj.execute('''insert into INSTRUCTOR values (2, 'Raul', 'Chong', 'Markham', 'CA'), (3, 'Hima', 'Vasudevan', 'Chicago', 'US')''') |

Để in dữ liệu trong bảng ra ta thực hiện như sau:

|  |
| --- |
| statement = '''SELECT \* FROM INSTRUCTOR'''  cursor\_obj.execute(statement)  print("All the data")  output\_all = cursor\_obj.fetchall()  for row\_all in output\_all:  print(row\_all) |

Nếu chỉ muốn in ra vài dòng thì thay đổi câu lệnh output\_all thành

|  |
| --- |
| output\_many = cursor\_obj.fetchmany(2) |

Để thực hiện nhiều lệnh truy vấn khác, chỉ cần tạo lệnh để đưa vào cursor\_obj.excecute():

|  |
| --- |
| statement = '''SELECT FNAME FROM INSTRUCTOR'''  cursor\_obj.execute(statement)  output\_column = cursor\_obj.fetchall()  query\_update='''update INSTRUCTOR set CITY='MOOSETOWN' where FNAME="Rav"'''  cursor\_obj.execute(query\_update)  output1 = cursor\_obj.fetchmany(2) |

Có thể kết hợp với thư viện pandas để tạo thành dataframe.

Sau khi thực hiện xong các lệnh truy xuất, ta cần đóng kết nối để giải phóng tài nguyê, tránh việc lạm dụng tài nguyên không cần thiết:

|  |
| --- |
| conn.close() |

Để tối giản các cách truy vấn khi sử dụng sql với python, tìm hiểu thêm về “sql magic”.

Ví dụ:

|  |
| --- |
| %sql  select count(\*) from CHICAGO\_PUBLIC\_SCHOOLS\_DATA where "Elementary, Middle, or High School"='ES' |

|  |
| --- |
| %sql  select Name\_of\_School, Average\_Student\_Attendance from CHICAGO\_PUBLIC\_SCHOOLS\_DATA \  order by Average\_Student\_Attendance desc nulls last limit 10 |

# 5.5 SQL nâng cao

## a. View

View là đoạn lệnh truy vấn đã được viết sẵn và lưu bên trong cơ sở dữ liệu. Một View thì bao gồm 1 câu lệnh SELECT và khi chạy View thì sẽ thấy kết quả giống như khi mở 1 Table. Nó giống như một Table ảo. Bởi vì nó có thể tổng hợp dữ liệu từ nhiều Table để tạo thành 1 Table ảo.

View rất hữu dụng khi muốn cho nhiều người người truy cập ở các mức đỗ khác nhau. Cụ thể là:

* Hạn chế truy cập tới các Table cụ thể. Chỉ cho phép được xem qua View.
* Hạn chế truy cập vào vào Column của Table. Khi truy cập thông qua View bạn không thể biết được tên Column mà View đó truy cập vào.
* Liên kết các Column từ rất nhiều Table vào thành Table mới được thể hiện qua View.
* Trình bày các thông tin tổng hợp (VD: sử dụng funtion như COUNT, SUM, ...)

Cú pháp tạo view:

|  |
| --- |
| CREATE VIEW view\_name AS  SELECT column1, column2, ...  FROM table\_name  WHERE condition |

Ví dụ tạo một view có tên là “toDolist”:

|  |
| --- |
| CREATE VIEW ToDoList AS  SELECT Tasks.TaskName, Tasks.Description  FROM Status INNER JOIN Tasks  ON Status.StatusId = Tasks.StatusId  WHERE (Status.StatusId = 1) |

Sau này khi bạn cần xem view hay lấy dữ liệu từ view ra thì chỉ việc gọi SELECT với bảng có tên là “ToDoList”. Nếu muốn chỉnh sửa view chỉ cần dùng lệnh ALTER:

|  |
| --- |
| ALTER VIEW ToDoList AS  SELECT Tasks.TaskName, Tasks.Description  FROM Status INNER JOIN  Tasks ON Status.StatusId = Tasks.StatusId  WHERE (Status.StatusName = 'To Do') |

Để xóa view khi không còn cần sử dụng nữa, dùng lệnh DROP:

|  |
| --- |
| DROP VIEW ToDoList |

## b. Commit and rollback

Commit là duy trì các thay đổi trong cơ sở dũ liệu, với cú pháp áp dụng cho MySQL như sau:

|  |
| --- |
| DELIMITER //  CREATE PROCEDURE PROCEDURE\_NAME  BEGIN  COMMIT;  END //  DELIMITER ; |

Rollback là khi thực hiện 1 hoặc nhiều dòng lệnh, nếu một lệnh trong số các lệnh không thể thực hiện được thì tất cả các lệnh đã thực hiện phía trước lệnh đó đều bị hủy, quay trở lại trước khi thực hiện những dòng lệnh. Cú pháp trong MySQL:

|  |
| --- |
| DELIMITER //  CREATE PROCEDURE PROCEDURE\_NAME  BEGIN  ROLLBACK;  COMMIT;  END //  DELIMITER ; |

Ví dụ về Rollback và commit:

|  |
| --- |
| DELIMITER //  CREATE PROCEDURE TRANSACTION\_ROSE()  BEGIN    DECLARE EXIT HANDLER FOR SQLEXCEPTION  BEGIN  ROLLBACK;  RESIGNAL;  END;    START TRANSACTION;  UPDATE BankAccounts  SET Balance = Balance-200  WHERE AccountName = 'Rose';    UPDATE BankAccounts  SET Balance = Balance+200  WHERE AccountName = 'Shoe Shop';    UPDATE ShoeShop  SET Stock = Stock-1  WHERE Product = 'Boots';    UPDATE BankAccounts  SET Balance = Balance-300  WHERE AccountName = 'Rose';      COMMIT;    END //  DELIMITER ; |

Để chạy đoạn code trên ta thực thi lệnh CALL với tên gọi của hàm:

|  |
| --- |
| CALL TRANSACTION\_ROSE; |

Tuy nhiên chỉ 1 trong 3 lệnh trong hàm TRANSACTION\_ROSE mà không thể thực hiện thì nó sẽ bị ROLLBACK quay trở về trước khi thực hiện lệnh CALL.

CHƯƠNG 6. Hands-on Introduction to Linux Commands and Shell Scripting

# 6.1 Giới thiệu về Linux, unix

Tham khảo thêm tại:

[Phân biệt giữa UNIX và Linux | Viết bởi steve\_jobs (tinhte.vn)](https://tinhte.vn/thread/phan-biet-giua-unix-va-linux.816063/)

# 6.2 Các lệnh trong Linux

|  |  |  |
| --- | --- | --- |
| Biểu tượng | Ý nghĩa | Ví dụ |
| ~ | Thư mục chính | Cd hoặc cd~ |
| / | Thư mục gốc |  |
| . | Thư mục làm việc hiện tại |  |
| .. | Thư mục cha của thư mục hiện tại | Cd ../ tăng lên 1 thư mục |

Một số lệnh trong Linux:

Whoami: trả về tên người dùng

Uname -a: trả về tên hđh, tên người dùng, …

Curl –help: nhận trợ giúp về các câu lệnh

Date: trả về ngày giờ hiện tại

Ls -lrt: liệt kê các thư mục theo ngày, từ mới nhất

Find -name \’\\*.sh\’: tìm thư mục có định dạng .sh

Pwd: trở về thư mục làm việc hiện tại

Mkdir newfolder: tạo thư mục mới

…

Cat file.txt: in ra các dòng trong file

Sort file.txt: sắp xếp và in ra

Sort -r file.txt: sắp xếp ngược và in ra

Uniq file.txt: loại bỏ các từ giống nhau

Grep ch file.txt: tìm các từ có chứa “ch” trong file

Grep -i ch file.txt: tìm nhưng ko phân biệt chữ hoa với chữ thường

Cut -c 2-9 file.txt: lấy ký tự thứ 2-9 trong mỗi dòng

Cut -d ‘ ‘ -f2 file.txt: lấy câu thứ 2 sau mỗi dấu cách

…

Hostname -i: trả về địa chỉ ip

Ifconfig: thông tin cấu hình thiết bị

Ifconfig tên mạng: trả về thông tin mạng

Curl [www.google.com](http://www.google.com): trả về html có chứa link hình ảnh GG, lấy hình ảnh bằng cách thêm [www.google.com/url](http://www.google.com/url)

Curl [www.google.com](http://www.google.com) -o google.text: lưu vào file text

Wget url: tải tệp từ url

Zip -r newname.zip file\_nén: nén file lại

Unzip name.zip: giải nén file

# 6.3 Tạo lịch tự động với crontab

Đầu tiên gõ vào command dong lệnh bên dưới để tạo file crontab mới.

|  |
| --- |
| crontab -e |

Sau đó sử dụng cấu trúc lệnh sau để tạo lịch:

|  |
| --- |
| m h dom mon dow command |

Trong đó: m là phút, h là giờ, dom là day of month (ngày trong tháng), mon là tháng, dow là day of week

Ví dụ:

|  |
| --- |
| 0 3 \* \* 1 tar -cvf my\_backup\_path\my\_archive.tar.gz $HOME\ |

Sao lưu thư mục chính vào thứ 2 hằng tuần vào lúc 3h sáng

Có thể sử dụng toán tử logic, vòng lặp, câu điều kiện với lệnh trong linux. Ví dụ như:

|  |
| --- |
| if [[ $# == 2 ]]  then  echo "number of arguments is equal to 2"  else  echo "number of arguments is not equal to 2"  fi |

|  |
| --- |
| for item in ${my\_array[@]}; do  echo $item  done |

CHƯƠNG 7. Relational Database Administration (DBA)

# 7.1 Tổng quan nhiệm vụ quản trị CSDL

Nền tảng của các khái niệm đạo đức cơ bản hỗ trợ thực hành bảo mật dữ liệu tốt. Những điều này sẽ giúp hướng dẫn các chính sách và quy trình làm việc cũng như các hành động thực hiện. Một số khái niệm quan trọng là:

* Tính minh bạch: Khi bạn thu thập thông tin, bạn nên cho chủ sở hữu thông tin biết chính xác dữ liệu nào bạn sẽ thu thập và bạn sẽ làm gì với dữ liệu đó. Thông báo cho họ về cách bạn sử dụng dữ liệu, cách bạn lưu trữ dữ liệu, ai sẽ có quyền truy cập vào dữ liệu đó và cách bạn sẽ xử lý dữ liệu khi bạn sử dụng xong.
* Sự đồng ý: Bạn nên nhận được sự đồng ý rõ ràng từ chủ sở hữu dữ liệu trước khi thu thập dữ liệu của họ. Điều này sẽ nêu chi tiết dữ liệu nào bạn sẽ được phép thu thập và cách bạn sẽ được phép sử dụng nó.
* Chính trực: Luôn rõ ràng về các thủ tục và chính sách của bạn, và luôn tuân theo chúng một cách nhất quán. Trong chừng mực bạn có thể, hãy đảm bảo rằng những người khác trong tổ chức của bạn cũng tuân theo các quy trình và chính sách chính xác.

Cấu trúc hệ thống là một công cụ mạnh mẽ trong việc giữ an toàn cho dữ liệu. Nếu hệ thống được xây dựng để duy trì bảo mật, việc ngăn chặn vi phạm sẽ dễ dàng hơn nhiều. Để đảm bảo hệ thống hoạt động, xem xét các yếu tố này.

* Bảo vệ khỏi truy cập độc hại: Tuyến đầu bảo vệ dữ liệu là bảo mật phần mềm cơ bản. Tường lửa và các công cụ an ninh mạng khác của bạn nên chủ động ngăn chặn hành vi tấn công và cài đặt phần mềm độc hại, đồng thời cảnh báo bạn về các mối đe dọa. Hãy chắc chắn rằng cập nhật phần mềm này thường xuyên, để giữ cho danh sách quét được cập nhật. Ngoài ra, giáo dục người dùng về lừa đảo và các cách khác mà họ có thể vô tình kích hoạt truy cập độc hại.
* Lưu trữ an toàn: Bộ lưu trữ được chọn cho dữ liệu của mình phải được bảo mật không chỉ khỏi truy cập độc hại mà còn khỏi lỗi phần cứng và thậm chí là thiên tai. Chọn dịch vụ một cách cẩn thận và đảm bảo hiểu các thực tiễn bảo mật và kế hoạch phòng chống thiên tai của họ. Sao lưu dữ liệu thường xuyên và đáng tin cậy để giảm thiểu mất dữ liệu trong trường hợp khẩn cấp.
* Truy cập chính xác: Chỉ những người cần một số dữ liệu nhất định mới có thể truy cập nó. Thiết lập một hệ thống gán và theo dõi các đặc quyền chỉ gán cho mỗi người dùng các đặc quyền cần thiết và kiểm soát những gì họ có thể làm với dữ liệu. Đảm bảo rằng chính sách của bạn tuân thủ mọi thỏa thuận sử dụng dữ liệu đã thực hiện.
* Di chuyển an toàn: Dữ liệu có thể đặc biệt dễ bị chặn khi di chuyển dữ liệu vào hoặc ra khỏi bộ nhớ. Hãy chắc chắn xem xét các phương pháp chuyển giao an toàn một cách cẩn thận như lập kế hoạch an toàn cho phần còn lại của hệ thống.
* Lưu trữ an toàn: Tại một số điểm, có thể muốn di chuyển dữ liệu từ bộ nhớ đang hoạt động sang kho lưu trữ. Điều này có thể bảo vệ nó khỏi truy cập ngẫu nhiên và làm cho hệ thống của bạn hiệu quả hơn. Đảm bảo hệ thống lưu trữ an toàn như phần còn lại của bộ nhớ. Thỏa thuận dữ liệu thường chỉ định thời gian bạn có thể sử dụng dữ liệu, vì vậy hãy đảm bảo dữ liệu được lưu trữ thường xuyên bị loại bỏ vì các quyền đã hết hạn và không giữ lại bất kỳ dữ liệu nào nhiều hơn mức cần để tuân thủ chính sách của tổ chức. Loại bỏ dữ liệu bị loại bỏ một cách an toàn và hoàn toàn.

# 7.2 Công cụ có sẵn trong MySQL

|  |  |
| --- | --- |
| Công cụ | Mô tả |
| InnoDB | * Công cụ lưu trữ mặc định cho MySQL 5.5 trở lên. * Thích hợp cho hầu hết các tình huống lưu trữ dữ liệu. * Cung cấp các bảng tuân thủ ACID và các ràng buộc về tính toàn vẹn tham chiếu FOREIGN KEY. * Hỗ trợ khả năng cam kết, khôi phục và khôi phục sự cố để bảo vệ dữ liệu. * Hỗ trợ khóa cấp hàng. * Lưu trữ dữ liệu trong các chỉ mục liên cụm giúp giảm I/O cho các truy vấn dựa trên khóa chính |
| MyISAM | * Quản lý các bảng phi giao dịch. * Cung cấp khả năng lưu trữ và truy xuất tốc độ cao. * Hỗ trợ tìm kiếm toàn văn bản. |
| MEMORY | * Cung cấp các bảng trong bộ nhớ, trước đây gọi là HEAP. * Lưu trữ tất cả dữ liệu trong RAM để truy cập nhanh hơn so với lưu trữ dữ liệu trên đĩa. * Hữu ích cho việc tra cứu nhanh tài liệu tham khảo và các dữ liệu giống hệt nhau. |
| MERGE | * Xử lý các nhóm bảng MyISAM tương tự như một bảng duy nhất. * Xử lý các bảng không giao dịch. |
| EXAMPLE | * Cho phép các nhà phát triển thực hành tạo một công cụ lưu trữ mới. * Cho phép các nhà phát triển tạo bảng. * Không lưu trữ hoặc tìm nạp dữ liệu. |
| ARCHIVE | * Lưu trữ một lượng lớn dữ liệu. * Không hỗ trợ chỉ mục. |
| CSV | * Lưu trữ dữ liệu ở định dạng Giá trị được Phân tách bằng Dấu phẩy trong tệp văn bản. |
| BLACKHOLE | * Chấp nhận dữ liệu để lưu trữ nhưng luôn trả về trống. |
| FEDERATED | * Lưu trữ dữ liệu trong cơ sở dữ liệu từ xa. |

Hiển thị thông tin trạng thái về các công cụ lưu trữ của máy chủ. Hữu ích để kiểm tra xem công cụ lưu trữ có được hỗ trợ hay công cụ mặc định là gì.

mysql> SHOW ENGINES;

# 7.3 Backup and restore

Có 3 loại sao lưu thông thường:

* Sao lưu đầy đủ: là sao lưu toàn bộ database, được thực hiện hằng tuần
* Sao lưu vi sai: là sự thay đổi nào đó xảy ra sau khi sao lưu đầy đủ, thực hiện mọi lúc
* Sao lưu gia tăng: bất kỳ nội dung xảy ra, được thực hiện mọi lúc

Sao lưu đầy đủ mất rất nhiều thời gian nên nó thường được thực hiện vào chủ nhật mỗi tuần, và khi đó dữ liệu không cập nhật nên gọi là sao lưu lạnh, còn sao lưu vi sai và gia tăng xảy ra khi người dùng đang làm việc, dữ liệu cập nhật liên tục gọi là sao lưu nóng.

Các hệ thống cơ sở dữ liệu khác nhau sử dụng thuật ngữ khác nhau cho nhật ký chứa thông tin giao dịch, ví dụ: MySQL gọi nó là nhật ký nhị phân, Postgres calls nó là nhật ký ghi trước và Db2 trên Cloud gọi nó là nhật ký giao dịch. Các file nhật ký thường được mã hóa nhị phân, nên cần có phần mềm chuyên dụng để đọc, ví dụ MySQL có mysqlbinlog để đọc, DB2 có db2fmtlog để đọc.

## Sao lưu trong MySQL

|  |
| --- |
| mysqldump --host=127.0.0.1 --port=3306 --user=root --password database\_name table\_name > file\_backup\_name.sql |

Thay thế database\_name, table\_name, file\_backup\_name thành cacsc tên cần thiết, sau khi ấn thì nhập mật khẩu của CSDL để tiến hành lưu file backup.

Tiếp theo để có thể sử dụng file backup cho MySQL ta thực hiện như sau:

|  |
| --- |
| mysql --host=127.0.0.1 --port=3306 --user=root --password database\_name < file\_backup\_name.sql |

Nhập mật khẩu để hoàn thành quá trình restore.

Để tránh những tai nạn xảy ra với CSDL cần phân quyền cho mỗi cá nhân, hạn chế những quyền hạn của mỗi người truy cập vào những thứ không cần thiết.

# 7.4 Tối ưu hóa cơ sở dữ liệu

Câu lệnh cung cấp thông tin về cách MySQL thực thi câu lệnh của bạn - nghĩa là cách MySQL dự định chạy truy vấn của bạn. Với bạn có thể kiểm tra xem truy vấn của bạn có đang lấy nhiều thông tin hơn mức cần thiết hay không, dẫn đến hiệu suất chậm hơn do xử lý lượng lớn dữ liệu bằng cách dùng lệnh EXPLAIN.

|  |
| --- |
| EXPLAIN SELECT \* FROM employees |

Khi có thể, tránh chọn tất cả các cột từ bảng của bạn. Với tập dữ liệu lớn hơn, việc chọn tất cả các cột và hiển thị chúng có thể mất nhiều thời gian hơn so với việc chọn một hoặc hai cột mà bạn cần.

Ví dụ: với tập dữ liệu khoảng 300.000 mục nhập của nhân viên, truy vấn sau mất khoảng 0,31 giây để tải. Nhưng nếu chúng ta chỉ muốn xem số lượng nhân viên và ngày tuyển dụng của họ (2 trong số 6 cột), chúng ta có thể dễ dàng làm như vậy với truy vấn mất 0,12 giây để tải này. Có thể nhận thấy thời gian truy vấn ngắn đi hơn 1/3.

Lưu ý: Ký tự đại diện ở đầu, là ký tự đại diện () tìm các giá trị kết thúc bằng các ký tự cụ thể, dẫn đến quét toàn bộ bảng, ngay cả khi có chỉ mục tại chỗ."%abc"

Tạo index cho các câu truy vấn, giúp cho truy vấn nhanh hơn nhưng cũng làm cho việc nhập dữ liệu vào chậm hơn. Khi tạo index thì sẽ đánh dấu số trang của mỗi đơn vị, khi truy xuất sẽ chia nửa index ra và tìm kiếm, tiếp tục chia nửa để tìm kiếm. Có thể tạo index như sau:

|  |
| --- |
| CREATE INDEX first\_name\_index ON employees(first\_name);  CREATE INDEX last\_name\_index ON employees(last\_name); |

Trong đó employees là tên bảng, first\_name, last\_name là các cột muốn tạo index.

Có thể sử dụng UNION ALL để làm giảm thời gian truy vấn trong một số trường hợp.

Một số phần mềm như của MySQL, DB2 có thể qua sát hiệu suất của CSDL thông qua trình hỗ trợ. Giúp người dùng có thể quan sát tốt hơn.

# 7.5 Khắc phục sự cố

Các vấn đề phổ biến nhất gặp phải với cơ sở dữ liệu thường do một hoặc nhiều nguyên nhân sau:

* Hiệu suất kém: có thể dẫn đến phản hồi chậm đối với các truy vấn của người dùng hoặc các ứng dụng truy cập cơ sở dữ liệu.
* Cấu hình không đúng: máy khách, máy chủ hoặc cơ sở dữ liệu được cấu hình không đúng cách có thể gây ra một loạt các vấn đề, bao gồm hiệu suất kém, sự cố, lỗi hoặc thậm chí hỏng cơ sở dữ liệu.
* Kết nối kém: có thể gây ra hiệu suất kém, hết thời gian chờ hoặc nhiều lỗi khi tương tác với cơ sở dữ liệu. Hiệu suất kém thường là do độ trễ đọc và ghi đĩa cao, thời gian xử lý chậm của máy chủ hoặc kết nối kém giữa máy chủ và máy khách.

Mỗi CSDL đều có cách để có thể xem được trạng thái của CSDL, ví dụ: trong Db2, bạn có thể chạy lệnh db2pd để theo dõi trạng thái của phiên bản Db2 và chạy chẩn đoán sự cố. Đối với cơ sở dữ liệu sử dụng MySQL, bạn có thể sử dụng lệnh HIỂN THỊ TRẠNG THÁI để lấy thông tin trạng thái máy chủ. Và trên máy chủ chạy PostgreSQL, bạn có thể sử dụng lệnh PG\_ISREADY để kiểm tra trạng thái kết nối của máy chủ cơ sở dữ liệu PostgreSQL, trên Microsoft SQL Server chạy trên Windows Server, bạn có thể sử dụng Giám sát Hoạt động để lấy thông tin về các quy trình SQL Server và cách các quy trình ảnh hưởng đến phiên bản hiện tại của SQL Server và sử dụng Giám sát Hệ thống để xác minh trạng thái và theo dõi hiệu suất SQL Server.

Khi xuất hiện một trạng thái lỗi thì đều đi kèm là một mã code, các trang web của CSDL có thể hỗ trợ thông tin mã code lỗi cũng như hướng giải quyết.

Để giải quyết các sự cố tốt nhất thì cần tạo một quá trình tự động hóa, khi sự kiên chỉ định đạt ngưỡng 85% thì cảnh báo và đạt đến ngưỡng 95% thì thông báo tự động bằng các phương thứ như SMS, email hoặc bảng điều khiển, mỗi trình duyệt web của CSDL đều hỗ trợ thông qua giám sát hiệu suất. Từ đó tạo quy trình tự động hóa cảnh báo lỗi.

CHƯƠNG 8. ETL and Data Pipelines with Shell, Airflow and Kafka

# 8.1 Mô tả ETL

Trích xuất, chuyển đổi và tải (ETL) là quá trình kết hợp dữ liệu từ nhiều nguồn thành một kho lưu trữ lớn trung tâm được gọi là kho dữ liệu. ETL sử dụng hệ thống các quy tắc kinh doanh để làm sạch và sắp xếp dữ liệu thô, sau đó chuẩn bị dữ liệu cho quá trình lưu trữ, phân tích và máy học (ML).

## Quy trình ETL

1. Trích xuất:

Trích xuất là bước đầu tiên trong cả ETL và ELT. Bước này thu thập dữ liệu thô từ các nguồn khác nhau. Những nguồn này có thể là cơ sở dữ liệu, tệp, ứng dụng phần mềm dưới dạng dịch vụ (SaaS), cảm biến Internet vạn vật (IoT) hoặc các sự kiện ứng dụng. Bạn có thể thu thập dữ liệu bán cấu trúc, có cấu trúc hoặc phi cấu trúc ở giai đoạn này.

1. Chuyển đồi:

Nhiều biến đổi khác có thể được áp dụng, bao gồm:

* Làm sạch: sửa mọi lỗi hoặc thiếu giá trị
* Lọc: chỉ chọn những gì cần thiết
* Tham gia: hợp nhất các nguồn dữ liệu khác nhau
* Chuẩn hóa: chuyển đổi dữ liệu sang các đơn vị phổ biến
* Cấu trúc dữ liệu: chuyển đổi định dạng dữ liệu này sang định dạng dữ liệu khác, chẳng hạn như JSON, XML hoặc CSV sang bảng cơ sở dữ liệu
* Ẩn danh và Mã hóa: đảm bảo quyền riêng tư và bảo mật
* Sắp xếp: sắp xếp thứ tự dữ liệu để cải thiện hiệu suất tìm kiếm

1. Tổng hợp: tóm tắt dữ liệu chi tiết Tải

Ở giai đoạn này, lưu trữ dữ liệu vào cơ sở dữ liệu

Khác với quy trình ETL (extra, transform, load) thì ở bước thứ 2 của ELT ko phải là transform mà là tải các nguồn dữ liệu vào kho lưu trữ trước, có thể cho vào data lake hoặc data market rồi sau đấy mới chuyển đổi dữ liệu.

## Các công cụ ETL phổ biến

* Talend Open Studio:

Hỗ trợ dữ liệu lớn, kho dữ liệu và lập hồ sơ

Bao gồm cộng tác, giám sát và lên lịch

GUI kéo và thả để tạo quy trình ETL

Tự động tạo mã Java

Tích hợp với nhiều kho dữ liệu

Mã nguồn mở

* AWS Glue

Dịch vụ ETL giúp đơn giản hóa việc chuẩn bị dữ liệu cho phân tích

Đề xuất lược đồ để lưu trữ dữ liệu của bạn

Tạo tác vụ ETL từ Bảng điều khiển AWS

* IBM InfoSphere DataStage

Một công cụ tích hợp dữ liệu để thiết kế, phát triển và chạy các tác vụ ETL và ELT

Thành phần tích hợp dữ liệu của IBM InfoSphere Information Server

Giao diện đồ họa kéo và thả

Sử dụng xử lý song song và kết nối doanh nghiệp trong một nền tảng có khả năng mở rộng cao

* Alteryx

Nền tảng phân tích dữ liệu tự phục vụ

Khả năng truy cập kéo và thả vào các công cụ ETL

Không cần SQL hoặc mã hóa để tạo quy trình

* Apache Airflow and Python

"Cấu hình" linh hoạt làm nền tảng đường ống dữ liệu mã

Mã nguồn mở bởi Airbnb

Tác giả, lên lịch và giám sát dòng công việc theo lập trình

Mở rộng quy mô thành dữ liệu lớn

Tích hợp với nền tảng đám mây

* The Pandas Python library

# Sử dụng command thực hiện ETL

|  |
| --- |
| echo "database" | cut -c1-4  echo "database" | cut -c5-8  echo "database" | cut -c1,5 |

Câu lệnh trên thự hiện cắt từ ký tự 1 đến 4, 5 đến 8 và ký tự 1 và 5 (char 1🡪4), nên output1 sẽ là “data”, output2 sẽ là “base”, output3 là “db”.

|  |
| --- |
| cut -d":" -f1 /etc/passwd  cut -d":" -f1,3,6 /etc/passwd  cut -d":" -f3-6 /etc/passwd |

Câu lệnh trên sẽ lấy ra các trường của tệp passwd

|  |
| --- |
| echo "Shell Scripting" | tr "[a-z]" "[A-Z]"  echo "Shell Scripting" | tr "[:lower:]" "[:upper:]" |

Câu lệnh trên sẽ chuyển đổi chữ viết thường thành viết hoa

|  |
| --- |
| ps | tr -s " " |

Thay thế các dấu cách trùng nhau thành 1 dấu cách

|  |
| --- |
| echo "My login pin is 5634" | tr -d "[:digit:]" |

Lệnh này xóa các kí tự sau -d, ví dụ “[:digit:] là xóa hết các ký tự số.

Trên là ý nghĩa của các câu lệnh cut và các -c,-d,-f,-s.

Để bắt đầu ta tạo 1 CSDL để chứa thông tin sau khi qua ETL, 1 file có bảng chứa trường thứ nhất là user name, trường thứ 3 là user id, trường thứ 6 là home directory, 1 file có tên “csv2db.sh”. Các câu lệnh thực hiện được viết vào file csv2db.sh và nguồn dữ liệu là /etc/passwd.

Trong file csv2db.sh gồm 3 quá trình.

Quá trình EXTRACT:

|  |
| --- |
| # Extract phase  echo "Extracting data"  # Extract the columns 1 (user name), 2 (user id) and  # 6 (home directory path) from /etc/passwd  cut -d":" -f1,3,6 /etc/passwd > extracted-data.txt |

Câu lệnh sẽ thực hiện lấy dữ liệu ở nguồn etc/passwd, các trường được ngăn cách nhau bằng dấu “:” và lưu vào file extracted-data.txt

Quá trình TRANSFORM:

|  |
| --- |
| # Transform phase  echo "Transforming data"  # read the extracted data and replace the colons with commas.  tr ":" "," < extracted-data.txt > transformed-data.csv |

Lấy dữ liệu từ file extracted-data.txt, chuyển đổi dấu ngăn cách “:” thành “,” và lưu vào file transformed-data.csv

Quá trình LOAD:

Để có thể tải dữ liệu vào kho PostgreSQL ta sử dụng cấu trúc sau:

|  |
| --- |
| COPY table\_name FROM 'filename' DELIMITERS 'delimiter\_character' FORMAT; |

Và file csv2db.sh được viết như sau:

|  |
| --- |
| # Load phase  echo "Loading data"  # Send the instructions to connect to 'template1' and  # copy the file to the table 'users' through command pipeline.  echo "\c template1;\COPY users FROM '/home/project/transformed-data.csv' DELIMITERS ',' CSV;" | psql --username=postgres --host=localhost |

Tổng hợp cả 3 quá trình vào file csv2db.sh để hoàn chỉnh file.

Khi chạy file chỉ cần mở command lên và gõ: “bash csv2db.sh”. Có thể chạy từng quá trình để test.

# Sử dụng Airflow để thực hiện ETL

Import các thư viện cần thiết:

|  |
| --- |
| # import the libraries  from datetime import timedelta  # The DAG object; we'll need this to instantiate a DAG  from airflow import DAG  # Operators; we need this to write tasks!  from airflow.operators.bash\_operator import BashOperator  # This makes scheduling easy  from airflow.utils.dates import days\_ago |

Định nghĩa AUG, có thể sử dụng cài đặt mặc định như sau:

|  |
| --- |
| #defining DAG arguments  # You can override them on a per-task basis during operator initialization  default\_args = {  'owner': 'Ramesh Sannareddy',  'start\_date': days\_ago(0),  'email': ['ramesh@somemail.com'],  'email\_on\_failure': False,  'email\_on\_retry': False,  'retries': 1,  'retry\_delay': timedelta(minutes=5),  } |

Tạo lịch trình cho DAG chạy hằng ngày:

|  |
| --- |
| # defining the DAG  # define the DAG  dag = DAG(  'ETL\_Server\_Access\_Log\_Processing',  default\_args=default\_args,  description='My first DAG',  schedule\_interval=timedelta(days=1),  ) |

Download file:

|  |
| --- |
| # define the tasks  # define the task 'download'  download = BashOperator(  task\_id='download',  bash\_command='wget "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0250EN-SkillsNetwork/labs/Apache%20Airflow/Build%20a%20DAG%20using%20Airflow/web-server-access-log.txt"',  dag=dag,  ) |

Tạo extract:

|  |
| --- |
| # define the task 'extract'  extract = BashOperator(  task\_id='extract',  bash\_command='cut -f1,4 -d"#" web-server-access-log.txt > /home/project/airflow/dags/extracted.txt',  dag=dag,  ) |

Transform:

|  |
| --- |
| # define the task 'transform'  transform = BashOperator(  task\_id='transform',  bash\_command='tr "[a-z]" "[A-Z]" < /home/project/airflow/dags/extracted.txt > /home/project/airflow/dags/capitalized.txt',  dag=dag,  ) |

Load:

|  |
| --- |
| # define the task 'load'  load = BashOperator(  task\_id='load',  bash\_command='zip log.zip capitalized.txt' ,  dag=dag,  ) |

Tạo quy trình:

|  |
| --- |
| # task pipeline  download >> extract >> transform >> load |

Tải DAG lên Airflow

|  |
| --- |
| cp ETL\_Server\_Access\_Log\_Processing.py $AIRFLOW\_HOME/dags |

Hoàn thiện file bằng cách copy các block code vào file “my\_first\_dag.py”. Sau đó thực hiện từng lệnh 1 vào command để chạy (Trước tiên phải chạy airflow rồi thự hiện các lệnh sau).

|  |
| --- |
| cp my\_first\_dag.py $AIRFLOW\_HOME/dags  cp my\_first\_dag.sh $AIRFLOW\_HOME/dags  cd airflow/dags  chmod 777 my\_first\_dag.sh  airflow dags list  airflow dags list|grep "my-first-dag"  airflow tasks list my-first-dag |

Để có thể tạm dừng, xem bản đồ hiệu suất, công việc một cách chính xác và đầy đủ hơn có thể xem tại trang chủ của airflow.

# Sử dụng Apache Kafka để xây dựng Pipelines for Streaming Data

Để có thể sử dụng Apache Kafka thì cần tải về và giải nén. Mở command lên và chạy các câu lệnh sau để tải kafka và giải nén:

|  |
| --- |
| wget <https://archive.apache.org/dist/kafka/2.8.0/kafka_2.12-2.8.0.tgz>  tar -xzf kafka\_2.12-2.8.0.tgz |

Tiếp theo khởi động Zookeeper dẻ kafka chạy được:

|  |
| --- |
| cd kafka\_2.12-2.8.0  bin/zookeeper-server-start.sh config/zookeeper.properties |

Mở terminal mới và chạy lệnh:

|  |
| --- |
| cd kafka\_2.12-2.8.0  bin/kafka-server-start.sh config/server.properties |

Mở terminal mới để tạo topic:

|  |
| --- |
| cd kafka\_2.12-2.8.0  bin/kafka-topics.sh --create --topic news --bootstrap-server localhost:9092 |

Sau đó chạy lệnh này để có thể tạo message:

|  |
| --- |
| bin/kafka-console-producer.sh --topic news --bootstrap-server localhost:9092 |

Tiếp theo sẽ có mũi tên hiện ra để có thể nhập message tùy ý.

Để khởi động, nhận được lời nhắn vừa tạo thì mở terminal mới và chạy dòng lệnh:

|  |
| --- |
| cd kafka\_2.12-2.8.0  bin/kafka-console-consumer.sh --topic news --from-beginning --bootstrap-server localhost:9092 |

Sau khi thực hiện xong chạy lệnh này để xóa các file kafka:

|  |
| --- |
| rm kafka\_2.12-2.8.0.tgz |

CHƯƠNG 9. Getting Started with Data Warehousing and BI Analytics

# 9.1 Tổng quan về kho dữ liệu

Kho dữ liệu là một hệ thống tổng hợp dữ liệu từ một hoặc nhiều nguồn thành một kho dữ liệu duy nhất, tập trung, nhất quán để hỗ trợ các yêu cầu phân tích dữ liệu khác nhau. Kho dữ liệu là một phần của hầu hết mọi ngành, bao gồm thương mại điện tử, vận tải, y tế, ngân hàng và công nghệ tài chính, truyền thông xã hội và chính phủ. Kho dữ liệu hỗ trợ khai thác dữ liệu, AI và học máy, OLAP và báo cáo front-end. Và cuối cùng, kho dữ liệu và BI giúp các tổ chức cải thiện chất lượng dữ liệu, tăng tốc độ hiểu biết kinh doanh và cải thiện việc ra quyết định, tất cả đều có thể dẫn đến lợi ích cạnh tranh.

Một số kho dữ liệu phổ biến như Oracle Exadata. Oracle Exadata có các thuật toán tích hợp và chạy tất cả các loại khối lượng công việc, bao gồm OLTP, phân tích kho dữ liệu, phân tích trong bộ nhớ và khối lượng công việc hỗn hợp. IBM Netezza được công nhận rộng rãi về khoa học dữ liệu và hỗ trợ học máy. Amazon RedShift sử dụng phần cứng dành riêng cho Amazon Web Services và phần mềm độc quyền trên đám mây để tăng tốc nén và mã hóa dữ liệu, machine learning và các thuật toán tối ưu hóa đồ thị tự động sắp xếp và lưu trữ dữ liệu. Tiếp theo, Google BigQuery mô tả hệ thống kho dữ liệu của mình là một "giải pháp kho dữ liệu đa đám mây linh hoạt". Google báo cáo thời gian hoạt động của kho dữ liệu là 99,99% và phân phối thời gian phản hồi truy vấn dưới giây từ bất kỳ công cụ nghiệp vụ thông minh nào. Microsoft Azure Synapse Analytics cung cấp các quy trình ETL/ELT trực quan không cần mã để nhập dữ liệu từ hơn 95 trình kết nối gốc. Azure Synapse Analytics hỗ trợ các trường hợp sử dụng kho dữ liệu và kho dữ liệu, đồng thời hỗ trợ sử dụng T-SQL, Python, Scala, Spark SQL và .Net cho cả tài nguyên phi máy chủ và chuyên dụng.

Việc lựa chọn kho dữ liệu cần xác định các tính chất sau:

* Chi phí
* Khả năng tương thích, triển khai
* Kho dữ liệu có thể tồn tại ở đâu (địa lý, đám mây, ..)
* Kiểu dữ liệu được hỗ trợ và kiểu dữ liệu công ty đang dùng là gì
* Sự hỗ trợ của bên cung cấp kho dữ liệu
* Công nghệ
* Phương án backup, restore

# 9.2 Tổng quan về data mark, data lake

Data mark là kho dữ liệu chứa các thông tin liên quan, hữu ích đối với một nhóm đối tượng cụ thể.

Data lake là tập hợp các thông tin từ nhiều nguồn đổ vào, nó chứa tất cả các loại thông tin, như 1 cái hồ chứa. Có thể thông tin có ích, thông tin rác, thông tin đã hết hạn, …

# 9.3 Triển khai kho dữ liệu

Đầu tiên cần xác định các bảng và các cột có trong bảng đó. Từ đấy tạo bảng với các thuộc tính như khóa chính, khóa phụ.

Để tạo bảng trong posrgres, sử dụng lệnh:

|  |
| --- |
| createdb -h localhost -U postgres -p 5432 practice |

Hoặc sử dụng phần mềm pgAdmin để thao tác, với giao diện thân thiện, dễ dùng.

## 9.3.1 Các lệnh nhóm dữ liệu

Mệnh đề GROUP BY trong SQL SERVER cho phép sắp xếp các bản ghi của một truy vấn theo nhóm. Các nhóm được xác định bởi các cột mà bạn chỉ định tròn mệnh đề GROUP BY. Nếu muốn thêm bất kỳ cột hoặc biểu thức nào không được liệt kê trong mệnh đề GROUP BY, bạn phải sử dụng cột đó làm đầu vào của hàm tập hợp. Nếu không sẽ gặp lỗi vì không có gì đảm bảo rằng cột hoặc biểu thức đó sẽ trả về một giá trị duy nhất cho mỗi nhóm.

Mệnh đề GROUPING SET là một nhóm các cột mà bạn nhóm. Thông thường một truy vấn với một tập hợp xác định một tập hợp nhóm duy nhất. Ví dụ sử dụng trong postgres.

|  |
| --- |
| select year,category, sum(billedamount) as totalbilledamount  from "FactBilling"  left join "DimCustomer"  on "FactBilling".customerid = "DimCustomer".customerid  left join "DimMonth"  on "FactBilling".monthid="DimMonth".monthid  group by grouping sets(year,category); |

Mệnh đề CUBE tạo tất cả các nhóm có thể dựa trên dữ liệu được chỉ định trong mệnh đề CUBE. Ví dụ:

|  |
| --- |
| select year,category, sum(billedamount) as totalbilledamount  from "FactBilling"  left join "DimCustomer"  on "FactBilling".customerid = "DimCustomer".customerid  left join "DimMonth"  on "FactBilling".monthid="DimMonth".monthid  group by cube(year,category)  order by year, category; |

Và mệnh đề nhóm ROLLUP. Ví dụ:

|  |
| --- |
| select year,category, sum(billedamount) as totalbilledamount  from "FactBilling"  left join "DimCustomer"  on "FactBilling".customerid = "DimCustomer".customerid  left join "DimMonth"  on "FactBilling".monthid="DimMonth".monthid  group by rollup(year,category)  order by year, category; |

# 9.4 Công cụ phân tích và nghiệp vụ thông minh (Analytics and Business Intelligence -BI)

Các tổ chức có thể sử dụng chuyên môn tích hợp của các công cụ BI để phân tích dữ liệu và tìm các mẫu bằng cách sử dụng máy học dựa trên mô hình thống kê. Phân tích mô tả cung cấp cái nhìn sâu sắc về quá khứ. Phân tích dự đoán cung cấp cái nhìn sâu sắc về những gì có thể xảy ra trong tương lai. Ngược lại, phân tích đề xuất cung cấp cái nhìn sâu sắc về những hành động mà một tổ chức nên thực hiện để tạo ra một kết quả cụ thể trong tương lai. Các công cụ nghiệp vụ thông minh hoặc công cụ BI cho phép chuẩn bị dữ liệu, khai thác dữ liệu, quản lý dữ liệu và trực quan hóa dữ liệu hiện nay như:

IBM Cognos Analytics, tích hợp Watson Analytics tận dụng trí tuệ nhân tạo và xử lý ngôn ngữ tự nhiên của IBM, Microsoft Power BI, nổi tiếng với thông tin chi tiết an toàn, Tableau được biết đến với trực quan hóa mạnh mẽ, đám mây Oracle Analytics được biết đến với phân tích hội thoại, SAP Business Objects được biết đến với các phân tích thông minh và Tibco Spotfire cung cấp khả năng mở rộng cho các tổ chức nhỏ và lớn.

Có thể sử dụng IBM Cognos Analytics phiên bản web để có cái nhìn trực quan, và có các biểu đồ cụ thể

CHƯƠNG 10. Introduction to NoSQL Databases

# 10.1 Tổng quan về NoSQL

Một số đặc điểm chung.

* High Scalability: Gần như không có một giới hạn cho dữ liệu và người dùng trên hệ thống.
* High Availability: Do chấp nhận sự trùng lặp trong lưu trữ nên nếu một node (commodity machine) nào đó bị chết cũng không ảnh hưởng tới toàn bộ hệ thống.
* Atomicity: Độc lập data state trong các operation.
* Consistency: chấp nhận tính nhất quán yếu, có thể không thấy ngay được sự thay đổi mặc dù đã cập nhật dữ liệu.
* Durability: dữ liệu có thể tồn tại trong bộ nhớ máy tính nhưng đồng thời cũng được lưu trữ lại đĩa cứng.
* Deployment Flexibility: việc bổ sung thêm/loại bỏ các node, hệ thống sẽ tự động nhận biết để lưu trữ mà không cần phải can thiệp bằng tay. Hệ thống cũng không đòi hỏi cấu hình phần cứng mạnh, đồng nhất.
* Modeling flexibility: Key-Value pairs, Hierarchical data (dữ liệu cấu trúc), Graphs.
* Query Flexibility: Multi-Gets, Range queries (load một tập giá trị dựa vào một dãy các khóa

Khác với SQL, NoSQL là kiểu dữ liệu không có cấu trúc, bât kỳ loại dữ liệu nào không phải là dữ liệu có kiến trúc.

* Có 4 loại NoSQL đó là:
* Document databases: Đây là loại cơ sở dữ liệu tài liệu lưu trữ và truy xuất dữ liệu dưới dạng một cặp khóa giá trị. Các giá trị đa dạng chẳng hạn như số, chuỗi, đối tượng…
* Key – Value databases: NoSQL loại này khá đơn giản vì mỗi mục của nó chỉ chứa key và value.
* Wide-column stores: Dữ liệu ở NoSQL loại này được lưu trữ trong bảng, cột động và hàng.
* Graph databases: Dữ liệu ở Graph databases được lưu trữ trong các nút và các cạnh.

Việc di chuyển RDBMS sang NoSQL cần rất nhiều kiến thức cũng như áp dụng các kiến thức SQL và NoSQL.

# 10.2 Giới thiệu về MongoDB

MongoDB là một CSDL NoSQL dạng document, thay vì lưu trữ dữ liệu trong các bảng hàng hoặc cột như cơ sở dữ liệu SQL, mỗi bản ghi trong cơ sở dữ liệu MongoDB là một tài liệu và bạn lưu trữ dữ liệu theo cách phi quan hệ. Tài liệu là các mảng liên kết như các đối tượng JSON hoặc từ điển Python. Mỗi trường trong MongoDB là cặp key-value, tương tự như các cột trong CSDL quan hệ

Lợi ích chính đầu tiên của việc sử dụng MongoDB là tính linh hoạt của lược đồ của nó, MongoDB cũng hỗ trợ tìm kiếm dữ liệu theo trường, truy vấn theo miền, hoặc tìm kiếm các biểu thức thông thường. Các truy vấn được thực hiện để có thể trả về các trường cụ thể trong document. Trong MongoDB có thể đánh chỉ mục cho bất kỳ trường nào, để giúp cho việc tìm kiếm dễ dàng hơn. MongoDB luôn sẵn sàng cung cấp bộ các bản sao. Một bộ bản sao bao gồm hai hoặc nhiều thể hiện của MongoDB.

## 10.2.1 Thao tác CRUD

Để thao tác với CSDL MongoDB ta sử dụng lệnh db.

Ví dụ:

|  |
| --- |
| Db.database\_name.hàm  Db.student.insertmany(…)  Db.student.findOne({“email”:”nguyenducquan@gmail.com”})  Student.find({“lastname”: “John”})  db.languages.updateMany({"name":"python"},{$set:{"creator":"Guido van Rossum"}}) |

Để tạo index cho một trường, sử dụng lệnh sau:

|  |
| --- |
| db.bigdata.createIndex({"account\_no":1})  db.bigdata.getIndexes() |

Câu lệnh trên tạo ra 1 index có tên là account\_no và in ra các index đã được tạo.

## 10.2.2 Sử dụng MongoDB với Python

Đầu tiên cài đặt MongoDB với python:

|  |
| --- |
| python3 -m pip install pymongo |

Để khởi động sử dụng lệnh:

|  |
| --- |
| start\_mongo |

Để kết nối với MongoDB Server, trước tiên tạo 1 file “mongo\_connect.py” và copy mã sau vào file.

|  |
| --- |
| from pymongo import MongoClient  user = 'root'  password = 'MjQwOTgtcnNhbm5h' # CHANGE THIS TO THE PASSWORD YOU NOTED IN THE EARLIER EXCERCISE - 2  host='localhost'  #create the connection url  connecturl = "mongodb://{}:{}@{}:27017/?authSource=admin".format(user,password,host)  # connect to mongodb server  print("Connecting to mongodb server")  connection = MongoClient(connecturl)  # get database list  print("Getting list of databases")  dbs = connection.list\_database\_names()  # print the database names  for db in dbs:  print(db)  print("Closing the connection to the mongodb server")  # close the server connecton  connection.close() |

Lưu file lại và để kết nối thực hiện lệnh sau:

|  |
| --- |
| python3 mongo\_connect.py |

# 10.3 Giới thiệu về Canssadra

Cassandra ban đầu được tạo ra bởi Facebook. Sau đó nó đã được tặng cho Quỹ Apache và tháng 2 năm 2010 và được nâng cấp lên thành dự án hàng đầu của Apache. Các tính năng ưu việt của Cassandra bao gồm 9 điểm sau:

* Thích hợp để sử dụng thực tế
* Khả năng chịu lỗi cao
* Kiến trúc không có SPOF (một điểm gây tổn hại)
* Mức độ tự do kiểm soát nhất quán
* Mô hình dữ liệu phong phú
* Có thể tăng cường cải thiện thông lượng cho tuyến tính
* Tính khả dụng cao
* Hỗ trợ các ngôn ngữ khác nhau dưới dạng client code
* Dễ dàng nắm bắt trạng thái bên trong của máy chủ bằng JMX/Dễ giám sát

## 10.3.1 Thao tác CRUD

Để tạo một keyspace có tên trainning cho Canssadra ta sử dụng lệnh:

|  |
| --- |
| CREATE KEYSPACE training  WITH replication = {'class':'SimpleStrategy', 'replication\_factor': 3}; |

Để chỉnh sửa một KeySpace, sử dụng lệnh ALTER:

|  |
| --- |
| ALTER KEYSPACE training  WITH replication = {'class': 'NetworkTopologyStrategy'}; |

Tạo 1 bảng trong keyspace như sau:

|  |
| --- |
| use training;  CREATE TABLE movies(  movie\_id int PRIMARY KEY,  movie\_name text,  year\_of\_release int  ); |

Chèn dữ liệu vào bảng như sau:

|  |
| --- |
| INSERT into movies(  movie\_id, movie\_name, year\_of\_release)  VALUES (1,'Toy Story',1995); |

Update dữ liệu trong bảng:

|  |
| --- |
| UPDATE movies  SET year\_of\_release = 1996  WHERE movie\_id = 4; |

Xóa dữ liệu trong bảng:

|  |
| --- |
| DELETE from movies  WHERE movie\_id = 5; |

# 10.4 Giới thiệu về Cloudant

Cloudant là một dạng CSDL đám mây của IBM.

Nhấp vào cloud.ibm.com/resources để đăng ký và xem hướng dẫn sử dụng.

Một số ví dụ về cú pháp trong IBM:

|  |
| --- |
| curl -X PUT $CLOUDANTURL/diamonds/1 -d '{  "carat": 0.31,  "cut": "Ideal",  "color": "J",  "clarity": "SI2",  "depth": 62.2,  "table": 54,  "price": 339  }' |

Truy vấn dữ liệu:

|  |
| --- |
| curl -X POST $CLOUDANTURL/diamonds/\_find \  -H"Content-Type: application/json" \  -d'{  "selector":  {  "\_id":"1"  }  }' |

Tạo index:

|  |
| --- |
| curl -X POST $CLOUDANTURL/diamonds/\_index \  -H"Content-Type: application/json" \  -d'{  "index": {  "fields": ["price"]  }  }' |

CHƯƠNG 11. Introduction to Big Data with Spark and Hadoop

# 11.1 Tổng quan về Big Data

Dữ liệu nhỏ là dữ liệu có sẵn với số lượng hạn chế có thể dễ dàng giải thích bởi con người với ít hoặc không có xử lý kỹ thuật số. Điểm số thể thao và lịch trình thay đổi của nhân viên là một số ví dụ về Dữ liệu nhỏ. Nó được tích lũy từ từ và có thể hoặc không thể được cập nhật liên tục.

Ngược lại với dữ liệu nhỏ là dữ liệu lớn, còn được gọi là big data. Big Data là các tập dữ liệu có khối lượng lớn và phức tạp. Độ lớn đến mức các phần mềm xử lý dữ liệu truyền thống không có khả năng thu thập, quản lý và xử lý dữ liệu trong một khoảng thời gian hợp lý. Những tập dữ liệu lớn này có thể bao gồm các dữ liệu có cấu trúc, không có cấu trúc và bán cấu trúc, nó còn được thu thập liên tục và phát triển theo cấp số nhân trong một khoảng thời gian ngắn. Dữ liệu lớn có thể ở bất kỳ dạng nào bao gồm nhưng không giới hạn ở văn bản, hình ảnh, âm thanh và video.

Các công cụ sử dụng big data như apache hadoop, apache HDFS và apache spark. Các công cụ phân tích dữ liệu phổ biến như: Tableau, Palantir, SAS, Pentaho, Oracle, PowerBI, Business Objects và Hyperion.

Hệ sinh thái công cụ Big Data bao gồm sáu loại công cụ chính sau: công nghệ dữ liệu, phân tích và trực quan hóa, kinh doanh thông minh, nhà cung cấp đám mây, cơ sở dữ liệu NoSQL và công cụ lập trình.

# 11.2 Tổng quan về Hadoop

Hadoop là một Apache framework mã nguồn mở cho phép phát triển các ứng dụng phân tán (distributed processing) để lưu trữ và quản lý các tập dữ liệu lớn. Hadoop giải quyết vấn đề như sau:

* Xử lý và làm việc khối lượng dữ liệu khổng lồ tính bằng Petabyte.
* Xử lý trong môi trường phân tán, dữ liệu lưu trữ ở nhiều phần cứng khác nhau, yêu cầu xử lý đồng bộ
* Các lỗi xuất hiện thường xuyên.
* Băng thông giữa các phần cứng vật lý chứa dữ liệu phân tán có giới hạn.

## 11.2.1 Hadoop MapReduce

Là cách chia một vấn đề dữ liệu lớn hơn thành các đoạn nhỏ hơn và phân tán nó trên nhiều máy chủ. Mỗi máy chủ có 1 tập tài nguyên riêng và máy chủ xử lý dữ liệu trên cục bộ. Khi máy chủ xử lý xong dữ liệu, chúng sẽ gởi trở về máy chủ chính. JobTracker là một điểm yếu trong Hadoop Mapreduce. Nếu JobTracker bị lỗi thì mọi công việc liên quan sẽ bị ngắt quãng.

## 11.2.2 Hadoop Distributed File System (HDFS)

Hệ thống tệp phân tán là một hệ thống tệp được phân phối trên nhiều máy chủ tệp và cho phép các lập trình viên truy cập hoặc lưu trữ tệp từ bất kỳ mạng hoặc máy tính nào. HDFS là lớp lưu trữ của Hadoop. HDFS hoạt động bằng cách chia các tệp thành các khối, sau đó tạo bản sao của các khối và lưu trữ chúng trên các máy khác nhau. HDFS được xây dựng để truy cập dữ liệu phát trực tuyến một cách liền mạch. Trong trường hợp mất dữ liệu của một trong các máy tính, dữ liệu có thể được tìm thấy trên một máy tính khác và công việc vẫn tiếp tục.

## 11.2.3 Hive

Hive là một phần mềm kho dữ liệu trong Hadoop được thiết kế để đọc, ghi và quản lý các bộ dữ liệu và phân tích dữ liệu lớn và dạng bảng. Kho dữ liệu lưu trữ dữ liệu lịch sử từ nhiều nguồn khác nhau để bạn có thể phân tích và trích xuất thông tin chi tiết từ đó. Những thông tin chi tiết này được sử dụng để báo cáo và sử dụng ngôn ngữ truy vấn dựa trên SQL, ví dụ như: Hive, hoặc Hive QL. Hive phù hợp để phân tích dữ liệu tĩnh (RDBMS phù hợp phân tích dữ liệu theo thời gian thực)

## 11.2.4 Hbase

HBase được sử dụng cho các ứng dụng ghi nhiều, để lưu trữ một lượng lớn bộ dữ liệu, xử lý và cung cấp phân tích trong thời gian thực. Hbase hỗ trợ sao lưu cho các công việc của MapReduce

## 11.2.5 Sử dụng Hadoop với shell

Đầu tiên cần download hadoop:

|  |
| --- |
| curl https://dlcdn.apache.org/hadoop/common/hadoop-3.3.6/hadoop-3.3.6.tar.gz --output hadoop-3.3.6.tar.gz |

Giải nén và cd đến thư mục hadoop:

|  |
| --- |
| tar -xvf hadoop-3.3.6.tar.gz  cd hadoop-3.3.6 |

Tải file txt để thử nghiệm:

|  |
| --- |
| curl https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-BD0225EN-SkillsNetwork/labs/data/data.txt --output data.txt |

Chạy câu lệnh này để đếm số từ xuất hiện trong tệp data.txt:

|  |
| --- |
| bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.6.jar wordcount data.txt output |

Để xem output của câu lệnh, chạy lệnh sau:

|  |
| --- |
| ls output |

Màn hình sẻ hiển thị tên file chứa output.

# 11.3 Apache Spark

Apache Spark là một framework dùng để xử lý dữ liệu, được xây dựng bằng ngôn ngữ scala.

Cung cấp các API cho Scala, Pytjom. Java. Tốc độ xử lý của Spark nhanh hơn gấp 10 lần MapReduce.

Những ưu điểm của Spark là:

* Xử lý dữ liệu: Spark xử lý dữ liệu theo lô và thời gian thực
* Tính tương thích: Có thể tích hợp với tất cả các nguồn dữ liệu và định dạng tệp được hỗ trợ bởi cụm Hadoop.
* Hỗ trợ ngôn ngữ: hỗ trợ Java, Scala, Python và R.
* Phân tích thời gian thực:

+ Apache Spark có thể xử lý dữ liệu thời gian thực tức là dữ liệu đến từ các luồng sự kiện thời gian thực với tốc độ hàng triệu sự kiện mỗi giây. Ví dụ: Data Twitter chẳng hạn hoặc luợt chia sẻ, đăng bài trên Facebook. Sức mạnh Spark là khả năng xử lý luồng trực tiếp hiệu quả.

+ Apache Spark có thể được sử dụng để xử lý phát hiện gian lận trong khi thực hiện các giao dịch ngân hàng. Đó là bởi vì, tất cả các khoản thanh toán trực tuyến được thực hiện trong thời gian thực và chúng ta cần ngừng giao dịch gian lận trong khi quá trình thanh toán đang diễn ra.

* Mục tiêu sử dụng:

+ Xử lý dữ liệu nhanh và tương tác

+ Xử lý đồ thị

+ Công việc lặp đi lặp lại

+ Xử lý thời gian thực

+ joining Dataset

+ Machine Learning

Apache Spark là Framework thực thi dữ liệu dựa trên Hadoop HDFS. Apache Spark không thay thế cho Hadoop nhưng nó là một framework ứng dụng. Apache Spark tuy ra đời sau nhưng được nhiều người biết đến hơn Apache Hadoop vì khả năng xử lý hàng loạt và thời gian thực.

## 11.3.1 Sử dụng Spark với Python

Đầu tiền cần tải một số gói spark:

|  |
| --- |
| !pip install pyspark  !pip install findspark |

Import thư viện:

|  |
| --- |
| import findspark  findspark.init()  from pyspark import SparkContext, SparkConf  from pyspark.sql import SparkSession |

Tải file về:

|  |
| --- |
| !curl https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-BD0225EN-SkillsNetwork/labs/data/people.json >> people.json |

In ra thông tin trong file:

|  |
| --- |
| df = spark.read.json("people.json").cache()  df.show()  df.printSchema() |

Tạo creatView:

|  |
| --- |
| df.createTempView("people") |

Sử dụng SparkSQL để truy vấn data:

|  |
| --- |
| df.select("name").show()  df.select(df["name"]).show()  spark.sql("SELECT name FROM people").show()  df.filter(df["age"] > 21).show()  spark.sql("SELECT age, name FROM people WHERE age > 21").show()  df.groupBy("age").count().show()  spark.sql("SELECT age, COUNT(age) as count FROM people GROUP BY age").show() |

Các câu lệnh SparkSQL tương tự với SQL nên thân thiện với người dùng nếu đã từng tiếp xúc với SQL.

CHƯƠNG 12. Machine learning with Apache Spark

# 12.1 Giới thiệu về Machine Learning

Học máy là các thuật toán giúp chương trình máy tính có khả năng học hỏi và hoàn thành các nhiệm vụ cụ thể được giao, và có thể tự động tối ưu hóa bản thân chương trình.

Vòng đời của học máy là:

Đầu tiên xác định vấn đề hoặc nêu tình huống, tiếp theo chuyển sang thu thập dữ liệu, sau đó chuẩn bị dữ liệu, tiếp theo phát triển và đánh giá mô hình và cuối cùng là triển khai mô hình.

Có 2 phương pháp học tập trong học máy: học có giám sat và học không giám sát. Học không giám sát hiện đang được sử dụng nhiều và có kết quả tốt.

Để đánh giá mô hình học máy thì cần tính toán độ chính xác, ma trận nhầm lẫn, độ chính xác tích cực và recall.

## 12.1.1 Machine learning với pandas

Đầu tiên tải các gói cài đặt cần thiết và import thư viện:

|  |
| --- |
| !pip install pandas==1.3.4  !pip install scikit-learn==1.0.2  !pip install numpy==1.21.6  import pandas as pd  from sklearn.linear\_model import LinearRegression |

Sau đó tải file csv:

|  |
| --- |
| URL = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-BD0231EN-SkillsNetwork/datasets/mpg.csv"  df = pd.read\_csv(URL)  df.sample(5) |

Tạo 1 biểu đồ để có thể hình dung được mối quan hệ giữa horsepower và mpg

|  |
| --- |
| df.plot.scatter(x = "Horsepower", y = "MPG") |

Tiếp theo lựa chọn đối tượng cần dự đoán và các đặc trung mà mô hình học

|  |
| --- |
| target = df["MPG"]  features = df[["Horsepower","Weight"]] |

Tạo một mô hình hồi quy tuyến tính

|  |
| --- |
| lr = LinearRegression() |

Và huấn luyện mô hình:

|  |
| --- |
| lr.fit(features,target) |

Cuối cùng là đánh giá mô hình và đưa ra dự đoán:

|  |
| --- |
| lr.score(features,target) |

Điểm số càng tiệm cận đến 1 thì mô hình được xem là càng tốt. Tiếp theo dự đoán quãng đường oto đi được với 100 mã lực và nặng 2000

|  |
| --- |
| lr.predict([[100,2000]]) |

Thu được kết quả là: 29.3216098 miles

Kết luận: Từ bảng có chứa mã lực, quãng đường đi được, khối lượng của xe, mô hình đã dự đoán được quãng đường xe chạy dựa vào mã lực và khối lượng xe.