**REPORT WEEK 2**

Contents

[**1.** **ETL** 2](#_Toc201915313)

[**1.1** **Khái niệm cơ bản** 2](#_Toc201915314)

[**1.2** **Cách thức hoạt động của ETL** 2](#_Toc201915315)

[**1.3** **Trích xuất dữ liệu (Extract)** 2](#_Toc201915316)

[**1.4** **Chuyển đổi dữ liệu(Transform)** 3](#_Toc201915317)

[1.5 Tải dữ liệu(Load) 3](#_Toc201915318)

[**2.** **AIRFLOW cơ bản** 3](#_Toc201915319)

[**2.1** **Apache Airflow là gì** 3](#_Toc201915320)

[**2.2** **DAG** 3](#_Toc201915321)

[**2.3** **Task** 4](#_Toc201915322)

[**2.4** **Operator** 4](#_Toc201915323)

[**2.5** **Sensor** 4](#_Toc201915324)

[**2.6** **Quy trình hoạt động của Airflow** 4](#_Toc201915325)

[**2.7** **Thực hành** 5](#_Toc201915326)

[**3.** **Apache Spark – Pyspark** 5](#_Toc201915327)

[**3.1** **Apache Spark là gì** 5](#_Toc201915328)

[**3.2** **Kiến trúc của Apache Spark** 6](#_Toc201915329)

[**3.3** **RDD là gì** 6](#_Toc201915330)

[**3.4** **Thực hành** 7](#_Toc201915331)

[**4. SQL nâng cao** 7](#_Toc201915332)

[**4.1 Giới thiệu** 7](#_Toc201915333)

[**4.2 Các chủ đề SQL nâng cao** 7](#_Toc201915334)

[**4.3 Thực hành** 8](#_Toc201915335)

1. **ETL**
   1. **Khái niệm cơ bản**

ETL là viết tắt của Extract – Transform – Load (trích xuất – biến đổi – tải )

Một hệ thống ETL được thiết kế phù hợp sẽ trích xuất dữ liệu từ các hệ thống nguồn, thực thi các tiêu chuẩn về tính nhất quán và chất lượng dữ liệu, tuân thủ dữ liệu để các nguồn riêng biệt có thể được sử dụng cùng nhau và cuối cùng cung cấp dữ liệu ở định dạng sẵn sàng để các nhà phát triển ứng dụng có thể xây dựng ứng dụng và người dùng cuối có thể đưa ra quyết định.

* 1. **Cách thức hoạt động của ETL**

Trích xuất, chuyển đổi và tải (ETL) hoạt động bằng cách di chuyển dữ liệu từ hệ thống gốc đến hệ thống đích trong các chu kỳ định kỳ. Quy trình ETL được hoạt động theo ba bước:

* Trích xuất dữ liệu có liên quan từ cơ sở dữ liệu
* Chuyển đổi dữ liệu để phù hợp hơn cho việc phân tích
* Tải dữ liệu vào cơ sơ dữ liệu đích
  1. **Trích xuất dữ liệu (Extract)**

Trong quá trình trích xuất dữ liệu, ETL sẽ trích xuất hoặc sao chép dữ liệu thô từ nhiều nguồn và lưu trữ chúng trong một khu vực đệm(khu vực lưu trữ trung gian để lưu giữ tạm thời dữ liệu đã trích xuất). Việc trích xuất dữ liệu thường diễn ra trong 1 trong 3 cách sau:

* Thông báo cập nhật: Trong quá trình thông báo cập nhật, hệ thống nguồn sẽ thông báo cho bạn khi bản ghi dữ liệu thay đổi. Sau đó, bạn có thể chạy quá trình trích xuất đối với thay đổi đó. Hầu hết các cơ sở dữ liệu và ứng dụng web đều cung cấp các cơ chế cập nhật để hỗ trợ phương pháp tích hợp dữ liệu này.
* Trích xuất tăng dần : Một số nguồn dữ liệu không thể gửi thông báo cập nhật nhưng có thể xác định và trích xuất dữ liệu đã được sửa đổi trong một khoảng thời gian nhất định. Trong trường hợp này, hệ thống kiểm tra các thay đổi theo định kỳ, chẳng hạn như mỗi tuần một lần, mỗi tháng một lần hoặc khi kết thúc chiến dịch. Bạn chỉ cần trích xuất dữ liệu đã thay đổi.
* Trích xuất hoàn toàn: Một số hệ thống không thể xác định các thay đổi dữ liệu hoặc gửi thông báo, do đó, tải lại tất cả dữ liệu là lựa chọn duy nhất. Phương pháp trích xuất này yêu cầu bạn giữ một bản sao của lần trích xuất cuối cùng để kiểm tra xem bản ghi nào là bản ghi mới. Bởi vì cách tiếp cận này có khối lượng truyền dữ liệu cao, chúng tôi khuyên bạn chỉ nên sử dụng cách này cho các bảng nhỏ.
  1. **Chuyển đổi dữ liệu(Transform)**

Trong chuyển đổi dữ liệu, ETL sẽ chuyển đổi và hợp nhất dữ liệu thô trong khu vực lưu đệm để chuẩn bị cho kho lưu trữ đích. Giai đoạn này thường liên quan dến các thay đổi dữ liệu sau:

* Chuyển đổi dữ liệu cơ bản: cải thiện chất liệu dữ liệu bằng cách loại bỏ lỗi, làm trống dữ liệu hoặc đơn giản dữ liệu bằng các cách như “Làm sạch dữ liệu”, “Chống trùng lặp dữ liệu”, “Sửa đổi định dạng dữ liệu”
* Chuyển đổi dữ liệu nâng cao: là việc sử dụng các quy tắc kinh doanh để tối ưu hóa dữ liệu nhằm phân tích dễ dàng hơn như “Dẫn xuất “, “Gộp ghép” , “Chia tách”’, “Tổng hợp”, “Mã hóa”
  1. Tải dữ liệu(Load)

Trong quá trình tải dữ liệu, ETL di chuyển dữ liệu đã chuyển đổi từ khu vực đệm vào khu vực đích. Đối với hầu hết các tổ chức quy trình này được tự động hóa, được xác định rõ ràng, liên tục và theo hang loạt có hai phương pháp để tải dữ liệu:

* Tải hoàn toàn : Toàn bộ dữ liệu được chuyển đổi và chuyển đến kho dữ liệu. Quá trình này thường diễn ra lần đầu tiên bạn tải dữ liệu từ hệ thống nguồn vào kho dữ liệu
* Tải tăng dần : Tải delta(hoặc chênh lệch) giữa hệ thống đích và nguồn theo những khoảng thời gian đều đặn Công cụ này sẽ lưu lại ngày trích xuất cuối cùng nhằm đảm bảo rằng chỉ tải các bản ghi được thêm vào sau ngày này. Có hai cách để thực hiện tải tăng dần là “Tải tăng dần theo luồng” hoặc “Tải tăng dần theo hàng loạt”

1. **AIRFLOW cơ bản**
   1. **Apache Airflow là gì**

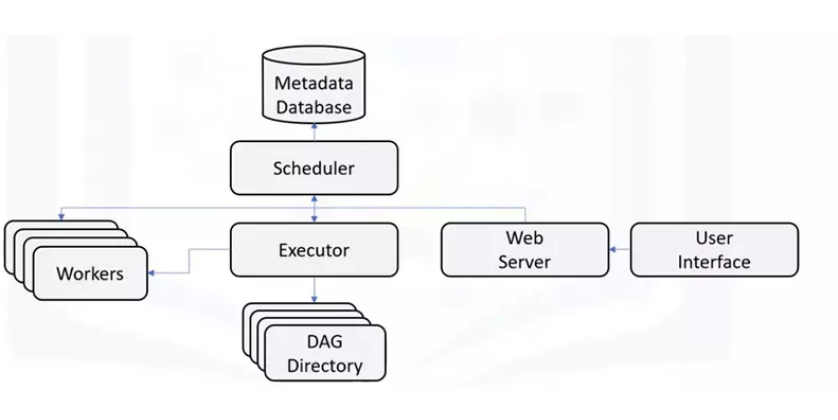
* Apache Airflow là một công cụ mã nguồn mở được sử dụng để “lập lịch”, “quản lý” và “ giám sát các quy trình xử lý dữ liệu”. Nó được sử dụng rộng rãi trong các hệ thống xử lý dữ liệu lớn để tự động hóa các quy trình xử lý dữ liệu phức tạp.
* Airflow cung cấp các khái niệm như “DAG”, “Task”, “Operator”, “Sensor” để mô tả các quy trình xử lý dữ liệu
  1. **DAG**
* DAG viết tắt của Director Acyclic Graph là một đồ thị có hướng không chu trình, mô tả tất cả các bước xử lý dữ liệu trong một quy trình
* Mỗi DAG được xác định trong 1 file DAG,nó định nghĩa một quy trình xử lý dữ liệu, được biểu diễn dưới dạng một đồ thị có hướng không có chu trình , trong đó các nút là các tác vụ(Tasks) và các phụ thuộc giữa các tác vụ
* Các tác vụ trong DAG thường được xử lý “tuần tự” hoặc “song song” theo một lịch trình có sẵn. Khi một DAG thực thi, nó được gọi là một lần chạy DAG
  1. **Task**

Task là một đơn vị cơ bản để thực hiện một công việc nhỏ trong quy trình xử lý dữ liệu. Mỗi task là một bước quy trình có thể lập lịch để thực hiện tùy theo các điều kiện cụ thể

* 1. **Operator**

Mỗi operator đại diện cho một công việc cụ thể trong quy trình, ví dụ như đọc dữ liệu từ một nguồn dữ liệu, xử lý dữ liệu, hoặc ghi dữ liệu vào một nguồn dữ liệu khác.

* 1. **Sensor**
* Sensor là một loại Operator được sử dụng để giám sát các sự kiện và điều kiện, và thực hiện các hành động tương ứng.
* Sensor thường được sử dụng để đợi cho đến khi một điều kiện nào đó xảy ra trước khi tiếp tục thực hiện quy trình.
  1. **Quy trình hoạt động của Airflow**
* Không giống như các công cụ Dữ liệu lớn như Apache Kafka, Apache Storm, Apache Spark,hoặc Flink, Apache Airflow không phải là giải pháp truyền dữ liệu. Nó chủ yếu là một trình quản lý quy trình làm việc

****

* Hình vẽ trên tổng quan về các thành phần cơ bản của Apache Airflow:
* Scheduler: giám sát tất cả DAG và các tác vụ được liên kết với chúng
* Executor: xử lý việc chạy các tác vụ này bằng cách đưa chúng cho worker để chạy
* Web server: giao diện người dùng của Airflow, hiện thị trạng thái của nhiệm vụ và cho phép người dùng tương tác với cơ sở dữ liệu
* DAG Directory: một thư mục chứa các file DAG của các quy trình xử lý dữ liệu (data pipelines)
* Metabase Database: được sử dụng bới Scheduler, Execulor và Web server để lưu trữ thông tin quan trọng của từng DAG

## **Thực hành**

* Đã thử dùng cài đặt Airflow bằng pip và chạy cơ bản

1. **Apache Spark – Pyspark**
   1. **Apache Spark là gì**

* Apache Spark là framework mã nguồn mở tính toán cụm được thiết kế để cung cấp tốc độ tính toán, khả năng mở rộng (Scalability) và khả năng lập trình cho bigdata. Đặc biệt, trong quá trình truyền một dữ liệu liên tục (Data streaming) với tốc độ cao, cơ sở dữ liệu đồ thị(Graph data), máy học(machine learning) và ứng dụng trí tuệ nhân tạo(AI)
* Apache Spark là engine tính toán đa dụng, nhanh và phân tán, có khả năng xử lý được lượng dữ liệu khổng lồ song song và phân tán trên nhiều máy tính đồng thời.
* Các thành phần của Apache Spark:
* Spark Core: là thành phần cốt lõi, các thành phần khác muốn hoạt động đều qua thông qua sparkcore, cung cấp các chức năng cơ bản để xử lý dữ liệu phân tán
* Spark SQL: cho phép người dùng truy vấn dữ liệu bằng ngôn ngữ SQL trên các tập dữ liệu lớn
* Spark Streaming: là một module cho phép xử lý dữ liệu theo thời gian thực
* Spark MLlib: là thư viện Machine Learning được tích hợp sắn trong Apache Spark, cung cấp các thuật toán Machine Learning phổ biến giúp ích trong việc xử lý bigdata
* GraphX: là đi kèm với lựa chọn các thuật toán phân tán để xử lý cấu trúc đồ thị

## **3.2 Kiến trúc của Apache Spark**

Đây là sơ đồ mô tả hoạt động của một ứng dụng Spark khi chạy trên 1 cụm máy tính :

A diagram of a cluster manager

AI-generated content may be incorrect.

* Driver Program: Trình điều khiển là tiến trình chính(main) chạy trên Master node có nhiệm vụ điều khiển, lập lịch điều phối công việc trong toàn bộ chương trình. Là tiến trình được khởi tạo đầu tiên và kết thúc cuối cùng ;
* Spark context: Là một đối tượng được khởi tạo bởi Driver program nằm trên master node, lưu trữ toàn bộ các thông tin về cấu trúc dữ liệu, cấu hình, kết nỗi đến Cluster manager.
* Cluster Manager: Quản lý và điểu phối tài nguyên trong cụm, Spark context Clutster manage có thể làm với nhiều loại khác nhau như Standalone, Apache Mesos….
* Executor : là các tiến trình chạy trên worker node , tiếp nhận và xử lý các task từ Driver
  1. **RDD là gì**
* RDD là một cấu trúc dữ liệu viết tắt của “Resillient Distributed Dataset” với các đặc điểm sau:
* Distributed: RDD có dạng collection của các Element, được chia nhỏ thành các partition và phân tán trên các node của cụm, từ cho phép dữ liệu RDD có thể xử lý song song
* Read only: RDD chỉ có thể được tạo từ dữ liệu input hoặc từ một RDD khác và không thể thay đổi sau khi tạo
* Persist: Spark duy trì RDD trên bộ nhớ trong suốt quá trình chạy, cho phép chúng có thể được tái sử dụng nhiều lần
* Fault tolerance: Cuối cùng RDD cũng có khả năng tự sửa chữa khi có node bị lỗi nhờ cơ chế tái tạo lại từ dữ liệu input hoặc các RDD khác
* RDD cung cấp 2 kiểu Operation là :
* Transformations: là các nhóm Operation tạo ra một RDD mới từ RDD đang có, ví dụ: map, flatmap…
* Actions: là nhóm các Operation cần trả về dữ liệu cho Drive sau khi thực hiên các tính toán trên RDD
  1. **Thực hành**
* Em đã cài đặt thành công Apache spark trên lap và test thử

**4. SQL nâng cao**

**4.1 Giới thiệu**

SQL nâng cao mở rộng khả năng xử lý dữ liệu của người dùng thông qua các tính năng mạnh mẽ như subquery, CTE, …. Đây là nền tảng để xây dựng các hệ thống dữ liệu phức tạp và hiêu quả

## **4.2 Các chủ đề SQL nâng cao**

* Subquery (Truy vẫn lồng nhau)
* Common Table Expressions (CTE)
* Window Functions (Hàm cửa sổ)
* Stored Procedures (Thủ tục lưu trữ)
* Trigger (Kính hoạt tự động)
* Transaction và Isolation Levels (Giao dịch và mức cô lập)
* Indexing và Query Optimization (Tối ưu truy vấn)

## **4.3 Thực hành**

* Em đã thực hành một số bài tập SQL trên Leetcode