1. Role of Data Engineering

1.1 Modern Data Ecosystem

Data Ecosystem은 상호연결적이고, 독립적이며, 끊임없이 진화하는 개체(Entity)들의 전체적인 네트워킹이다.

1.2 Data Source

데이터는 정형, 비정형 등 다양한 데이터 타입으로 존재할 수 있다. 더불어 텍스트, 이미지, 비디오, 클릭 스트림, 유저간의 대화, IoT, 실시간 스트림 데이터, 전문적인 기관에서 제공하는 Data Source가 존재한다.

가장 첫번째로 해야할 일은 데이터 저장소(Data Repository)나 데이터 소스(Data Source)에서 데이터를 추출하는 것이다. 이 단계에서 우리가 작업하고자 하는 데이터의 형태, 소스, 인터페이스에 따른 데이터를 추출하는 방식을 확인한다.

한번 Raw Data를 추출하게 되면, 이 데이터는 최종 사용자를 위해 정돈되고, 최적화될 필요성이 있다. 따라서 데이터 관리(Data Management)는 높은 수준의 가용성, 유용성, 접근성, 안전성을 제공하는 데이터 저장소(Data Repository)를 구축하는 작업을 수행하게 된다.

마지막으로 Business Shareholders, Applications, Analyst, Programmer, Data Scientist/Engineer/Analyst 등 사용자에 맞게 데이터에서 적절한 데이터를 제공해주면 된다.

1.3 Key Players in the Data Ecosystem

데이터 엔지니어(Data Engineer)는 데이터 구조를 유지하고 개발하며 비즈니스와 분석을 위해 데이터를 이용 가능하게 만드는 작업을 수행한다. 데이터 엔지니어는 서로 다른 데이터 소스에서 얻어진 데이터를 추출, 통합, 정돈하기 위해 데이터 에코시스템(Data Ecosystem)에서 작업하게 된다. 그들은 데이터가 다양한 시스템과 형테에 접근가능하도록 하며, 이를 위해 프로그래밍 스킬, 시스템과 CS지식, 관계형 데이터베이스와 비관계형 데이터베이스에 대한 지식이 필요하다.

1.4 What is Data Engineering?

데이터 엔지니어링 분야는 다른 소스에서 얻어진 데이터를 추출, 수집, 통합, 최적화하는 작업을 수행한다.

필요한 데이터를 얻기 위해서는 Developer tools, Workflow, 다른 데이터 소스에서 추출하는 프로세스가 필요하다. 이 데이터들을 데이터베이스, 데이터 웨어하우스(Data Warehouse), 데이터 레이크(Data Lake)등 데이터 저장소에 저장되게 된다.

데이터를 처리하는 과정은 정제, 변환, 정돈(ETL)을 수행하는 작업이다, 데이터 엔지니어는 Large-Scale 데이터를 처리학 위해 분산 시스템(Distributed-System)을 유지, 구현하게 된다. 더불에 데이터를 추출, 변환, 적재하는 파이프라인(Pipeline)을 설계하기도 한다. 데이터가 규제와 Compliance(사업 추진 과정에서 기업이 자발적으로 관련 법규를 준수하도록 하기 위한 일련의 시스템) 가이드라인을 준수하는지 검수하는 작업도 진행한다.

데이터 엔지니어는 이용가능한 데이터를 저장하는 작업을 수행하고 처리된 데이터를 저장하기 위한 데이터 스토어(Data Store)를 구현하고 구축한다. 또한, 데이터 에코시스템을 구성하는 Tools와 Systems들이 Privacy, Security, Compliances, Monitoring, Backup, Recovery를 수행하도록 구현한다.

또한, 최종 사용자(APIs, Services, Programs, Users)들이 데이터를 안전하게 이용할 수 있도록 데이터를 제공하며, 데이터에서 인사이트를 얻을 수 있도록 Interface와 DashBoard를 제공하기도 한다.

즉. 데이터 엔지니링은 분석과 의사결정을 위해 질 좋은 데이터를 만드는 것에 목표를 두고 있다. 그리고 이 목표는 원천 데이터 소스에서 데이터를 수집하고, 가공하며, 저장하고, 사용가능하게 만듦으로써 가능하다.

2. Responsibilities, Skillsets

2.1 Responsibilities of Data Engineering

데이터 엔지니어의 가장 중요한 책임감은 "데이터 소비자에게 이용 가능한 데이터를 제공한다"는 것이다. 따라서 일반적으로 다른 소스에서 온 데이터를 추출, 정돈, 통합하고, 데이터 파이프라인(Data Pipeline)을 설계 및 관리하며, 데이터 수집과 저장에 필요한 시스템을 설계하고 관리하게 된다(Data Platforms, Data Store, Distributed Systems, Data Repositories).

2.1 Technical Skilss for Data Engineer

- Operating Systems : UNIX, LINUX, Windows, System Utilities and Commands
- Infrastructure Components : Virtual Machine, Networking, Application Services
- Cloud-Based Services : Amazon, Google, IBM, Microsoft
- RDBMS : MySQL, Oracle Database, PostgreSQL
- NoSQL : Redis, MongoDB, Cassandra, Neo4J
- Data Warehouse : Oracle Exadata, Amazon RedShift
- Data Pipeline : Apache Beam, Apache Airflow, DataFlow
- ETL Tools : AWS Blue, Improvado
- Query Language : SQL, NoSQL
- Programming Languages : Python JAVA
- BigData Processing Tools : Hadoop, Hive, Spark

2.2 Functional Skills

데이터 엔지니어는 비즈니스 요구를 기술적 구체화로 변경할 수 있어야 하며, 데이터의 잠재적인 적용을 이해하고 있어야 한다. 특히, 나쁜 데이터 관리의 위험을 이해고 있어야 한다.(About data quality, privacy, security, compliance)

3. Data Ecosystem, Languages

3.1 Overview of the Data Engineering System

데이터 엔지니어링 에코시스템은 기반시설(Infrastructure), 툴(Tools), 프레임워크(Frameworks), 프로세스(Process)를 다음 작업을 위해 포함하고 있다.

- 다른 소스들로부터 데이터를 추출한다.
- 데이터 저장, 변환, 통합을 위한 데이터 파이프라인을 구축하고 관리한다.
- 데이터 저장소를 구축하고 관리한다.
- 시스템 사이에 워크플로우를 최적화하고 자동화한다.
- 데이터 엔지니어링 워크플로우를 위한 어플리케이션을 개발한다.

3.2 Types of Data

데이터는 정돈되지 않은 정보로, 의미를 갖기 위해서는 전처리가 필요하다. 일반적으로 데이터는 사실, 관찰, 인식, 숫자, 문자, 상징, 이미지로 구성되며 이를 이끌어내기 위해 존재한다.

Structured Data

정형 데이터(Structured Data)는 객관적인 사실과 숫자로, 전통적인 데이터베이스에서 수집, 송출, 저장, 정돈된 데이터이다. 정형 데이터를 포함하고 있는 데이터 소스는 다음과 같다.

- SQL Databases
- OLTP
- SpreadSheets(Excel, Google Spreadsheets)
- Sensor(GPS, RFID)
- Network, Webserver logs

정형 데이터는 관계형, SQL 데이터베이스에서 저장되며 표준적인 분석 방법과 툴로 다뤄질 수 있다.

Semi-Structured Data

반정형 데이터(Semi-Structured Data)는 어느 정도 조직화된 속성이 있으나 고정된 스키마가 부족하다. 반정형 데이터는 다음과 같은 포맷으로 존재한다.

- XML, JSON
- TCP/IP Packets
- Zipped Files

Unstructured Data

비정형 데이터(Unstructured Data)는 쉽게 식별되지 않는 구조를 가지고 있으며 관계형 데이터베이스에서 열과 행의 형태로 정돈될 수 없다. 비정형 데이터는 특징적인 형태, 순서, 규칙이 없으며 이질적인 소스와 다양한 BI, 분석 어플리케이션으로 수행된다.

- Web Pages
- Social Media Feeds
- Images
- Video and Audio Files
- Documents and PDF files

3.3 Understanding Different Types of File Formats

Delimited text files

Delimited text files은 데이터를 각각의 행(record)에 저장하기 위한 text file로 delimiter에 의해 분리되어 있다. 어떠한 문자도 값을 분리하는데 사용될 수 있지만, 흔히 사용되는 분리자(delimiter)는 comma(,), tab(), colon(;), 그리고 space(" "). Comma-Separated values (or CSVs) 와 Tab-separated values (or TSVs) 는 Delimiter text files에서 가장 흔히 쓰이는 형식이다.

Microsoft Excel Open XML or XLSX

Microsoft Excel open XML Spreadsheet는 Microsoft Excel Open XML 파일 형식으로 spreadsheet 파일 형태를 가지고 있다. XLSX에서는 여러 개의 worksheets를 사용할 수 있으며, 각각의 worksheet는 행과 열로 구성되어 있다.

Markup Language or .XML

Extensible Markup Language, 혹은 XML은 마크업(mark up)언어로 데이터를 입력하는 규칙을 가지고 있다. XML파일 형식은 인간과 기계가 동시에 해석이 가능하며, 정보를 인터넷으로 보낼때 self-descriptive한 언어이다.

XML은 HTML과 어느정도 유사하지만 HTML과는 다르다. XML은 HTML과는 다르게 미리 정의된 tag는 사용하지 않으며, 플랫폼과 프로그래밍 언어에 독립적이어서 시스템간 데이터 공유를 간단화할 수 있다.

PDF

Portable Document Format, 혹은 PDF은 Adobe에 의해 개발된 파일 형식이다. PDF파일은 소프트웨어, 하드웨어, 운영체제에 관계없이 사용이 가능하며 법적이고 금융적인 문서 형식에 사용된다.

JSON

Javascript Object Notation, 혹은 JSON는 text-based 언어로 정형화된 데이터를 웹으로 보내기 위해 사용된다. JSON은 프로그래밍 언어에 관계없이 읽을 수 있으며, 쉽게 사용이 가능하고, 어떤 데이터 타입에 대해서 공유가 쉬운 파일 형식이다.

3.4 Sources of Data

Relational Databases

조직은 그들의 Business Activities, Customer Transaction, Human Resource Activities 등 Workflow를 위해 조직 내부의 어플리케이션을 가지고 있다. 이러한 시스템은 관계형 데이터베이스를 사용하여 정형화된 방식으로 데이터를 저장한다,

Flatfiles and XML Datasets

조직 외부에서는 공식적이거나 개인적으로 이용가능한 데이터가 존재한다. 이러한 데이터들은 보통 CSV파일, XML문서로 제공이 된다. Flatfiles에 대표적으로 이용되는 파일 형식은 CSV로 Delimiter에 의해 본리되는 행과 열의 구조를 가지고 있다. XML은 mark up된 tabg를 이용하여 위계적인 데이터를 설명하는데 유용항 파일 형식이다.

APIs and Web Services

많은 데이터 공급자와 웹사이트는 APIs(Application Program Interface)와 Web Services를 제공한다. APIs는 복수의 유저와 어플리케이션들이 서로 상호작용하며 처리와 분석을 위한 데이터를 얻는데 사용된다. APIs와 Web Services는 들어오는 요구를 처리하고 그 데이터를 Plain text, XML, HTML, JSON 혹은 media 파일형식으로 내보낸다.

APIs는 또한 데이터베이스 소스로부터 데이터를 꺼내는데 사용되며, Web Scrapingweb은 비정형인 데이터를 추출하는데 사용된다.

Data Streams and Feeds

Data streams은 데이터가 지속해서 흐르는 기구, IoT장치, 어플리케이션, GPS, Website, Social media에서 데이터를 집계하는데 널리 사용되고 있다. 데이터 스트림을 처리하는데 있어 가장 많이 사용되는 어플리케이션은 Apache Kafca, Apache Spark Streaming, Apache Storm이 있다.

RSS feeds는 다른 데이터 소스인데, Online form이나 news website에서 업데이트 되는 데이터를 포착하는데 사용된다.

3.5 Languages for Data Professionals

Query Language

Query Languages는 데이터베이스에서 데이터를 조작하고 접근하는데 사용된다. SQL or Structured Query Language는 관계형 데이터베이스에서 데이터를 접근하고 조작하는 querying language이다. SQL을 사용함으로써 우리는 데이터베이스의 값들을 삭제, 삽입, 업데이트하는 명령을 작성할 수 있다.

Advantages of using SQL :

- SQL은 플랫폼에 독립적으로 사용가능하다.
- 넓은 폭과 특수한 확정으로 데이터를 질문할 수 있다.
- 영어와 비슷한 문법을 가지고 있다.
- 다른 프로그래밍 언어와 다르게 적은 줄로 프로그램을 작성할 수 있다.
- 많은 양의 데이터를 빠르고 효과적으로 찾을 수 있다.
- 소스코드를 바로 실행할 수 있다.

Programming Languages

프로그래밍 언어는 어플리케이션을 개발하고 통제하는데 사용된다. 대표적으로 Python, R, JAVA가 있다. Python은 넓게 사용되ㄴ Open-Source, High-level Programming Languae이다. Python의 문법은 프로그래머들이 개념을 적은 코드에 표현할 수 있으며 배우기 쉽고, 사용자가 가장 많은 언어이다.

Python은 Numpy와 Pandas와 같은 복잡한 계산을 병렬적으로 수행하는 Library를 제공한다.

Advantages of using Python :

- Easy to learn
- Open-source
- Various community offering various analysis libraries
- Scikit-learn, Scipy, Pytorch, ... for data science

JAVA는 객체지향적, 클래스 기반, 플랫폼 독립적인 언어이다. 빅데이터 시대에 가장 인기있는 프레임워크들과 툴들은 JAVA로 구성되어 있는데, 예를 들어 Hadoop, Hive, Spark가 있다.

Shell Scripting

Unix/Linux Shell이나 Powershell과 같은 Shell Scripting언어는 반복적이고 시간 소비적인 작업에 이상적이다. Shell Scripting 언어에 의해 수행되는 작업은 다음과 같다 : 파일조작, 프로그램 실행, 시스템 관리, 디스크 백업, 시스템 로그 평가, 프로그램 설치.

Powershell은 Microsoft에서 개발된 크로스 플랫폼 자동화 툴로 JSON, CSV, XML, REST APIs 등과 같은 정형 데이터를 다루는데 최적화가 되어있다. Powershell은 Command-Line Shell과 Scripting 언어로 구성되어 있다. 또한 객체 기반으로 요약, 측정, 그룹핑 등 여러가지 작업을 데이터 파이프라인 내부에서 사용가능하다.

3.6 Metadata and Metadata Management

메타데이터는 데이터에 대한 정보를 제공하는 데이터이다.

Technical Metadata

기술적 메타데이터는 데이터 저장소나 플랫폼에 있는 데이터 구조를 기술적인 관점에서 정의하는 데이터이다.

- Tables that record information about the tables stored in a database
    - each table's name
    - the number of columns and rows each table has
- A data catalog, which is an inventory of tables that contain information
    - the name of each database in the enterprise datawarehouse
    - the name of each column present in each database
    - the names of every tables that each column is contained in
    - the type of data that each column contains

Process Metadata

프로세스 메타데이터는 비즈니스 시스템 뒤에 있는 작동 과정에 대해서 묘사하는 데이터이다. 예를들어 데이터 웨어하우스, 회계 시스템, 고객 관리 장치 등 이 있다.

기업 시스템은 다양한 데이터 소스로부터 데이터를 모으고 처리하는데 책임이 있다. 이러한 순환 시스템은 실패와 비 정상적인 작동이 발생하는 순간을 모니터링 할 필요가 있다. 그래스 프로세스 데이터는 다음과 같은 작업을 추적하는 일을 수행한다.

- Process start and end times
- Disk usage
- Where data was move from and to
- How many users access the system at any given time

이러한 데이터는 TroubleShootting 이나 Workflow를 최적화하는데 사용된다.

Bussiness Metadata

데이터를 분석하고 탐색하는 유저는 데이터 발견에 관심이 있다. 그들은 의미있거나 귀중한 데이터를 찾을 필요가 있으며 데이터를 어디서 접근할 수 있는지 알아야 한다. 비즈니스 메타데이터는 데이터를 해석하기 쉽게 다음과 같은 정보를 제공한다.

- How the data is acquired
- What the data is measuring or describing
- The connection between data and other data sources

Why is the metadata management important?

좋은 메타데이터 관리는 가치있는 이익을 제공한다. 잘 구현된 데이터 카탈로그는 데이터 발견, 거버넌스, 접근을 쉽게 가능하게 한다. 또한 기업 데이터와 관련된 비즈니스 정보와 데이터 연계를 이해하기 쉽게 해주고 이를 통해 데이터 거버넌스를 향상시키는데 도움을 준다.

4. Data Responsibilities, Data Pipelines, Data Integration Platforms

4.1 What is data repositories?

데이터 저장소(Data Repository)는 데이터가 비즈니스 작업, 분석, 보고를 위해 수집되고, 정돈되며 구정된 장소이다. 데이터 저장소는 하나 이상의 데이터가 모여 작은 구조부터 큰 구조까지 다양하게 존재한다.

4.2 RDBMS

What is RDBMS

관계형 데이터베이스는 테이블 구조의 데이터가 모여있는 집합이다. 이 테이블들은 공통된 컬럼에 따라 서로 연결되고 결합되어 있다. 테이블들은 행과 열의 구조로 구성되어 있는데, 행은 Records, 열은 Attributes로 정의한다.

공통의 데이터에 기반하여 테이블을 연결하는 것은 우리가 한줄을 Query로 하나 이상의 새로운 테이블을 건색하게 할 수 있다. 이것은 모든 이용가능한 데이터들 사이에서 관계를 이해하고 새로운 통찰력으로 올바른 결정을 내릴 수 있게 끔 도와준다.

관계형 데이터베이스는 SQL(Structured Query Language)를 사용한다. 또한 정의된 구조와 Schema를 따르는 행과 열로 구성되어 있는 조직적인 원칙을 따른다.

Advantages of using RDBMS

- 많은 양의 데이터를 처리, 검색, 저장하는데 최적화되어 있다.
- 각각의 테이블은 행과 열로 구성되어 있다.
- 테이블 사이에 관계를 정의할 수 있다.
- 각 필드는 특정한 데이터 타입과 값으로 제한된다.
- SQL을 사용하면 수많은 레코드를 몇 초안에 검색할 수 있다.
- 관계형 데이터베이스의 안정적인 구조는 Governance와 Accessibility를 지원한다.

Services for RDBMS

- Relational Databases : IBM DB2, SQL Server, MySQL, Oracle DB, PostgreSQL
- Cloud-Based Relational Databases : Amazon RDS, Google SQL, IBM DB2 on Cloud, Oracle Cloud, AzureSQL

Usecase for RDBMS

- OLTP(Online Transacion Processing)
    - 높은 비율로 실행되는 Transaction-Oriented 작업을 지원한다.
    - 트랜잭션은 쪼갤 수 없는 업무의 최소 단위이다.
    - 작은 양의 데이터를 관리한다.
    - 빠른 응답시간과 빈도가 많은 Query를 지원한다.
- Data Warehouse
    - OLAP(Onlien Analytical Processing)을 위해 최적화되어 있다.
    - Data를 분석하고 의미있는 정보로 치환하거나, 복잡한 모델링을 가능하게끔 하는 분석 방법이다.

4.3 NoSQL

What is NoSQL

NoSQL은 Not Only SQL의 줄임말로, 데이터를 저장하고 검색하는데 유연한 Schema를 제공하는 비 관계형 데이터베이스이다. NoSQL은 사용하기 간단하고, 수행능력, 확장성과 관련된 속성에 의해 주목받으며, 특정한 데이터 모델을 위해 구성된다. 현대에 들어서 Application을 생성하고 관리하기 위한 유연한 Schema를 가지고 있다. 따라서 전통적인 행, 열, 테이블 구조의 데이터베이스를 가지고 있지 않다.

Common Types of NoSQL Databases

- Key-Value Store
    - Key-Value 쌍으로 데이터를 저장한다.
    - Key는 데이터의 속성을 나타낸다.
- Document Based Store
    - 각각의 Record에 단일 문서를 저장한다.
    - 강력하고 유연한 인덱싱과 비정형적인 Query가 가능하다.
- Column Based Store
    - 행과 그룹화된 컬럼을 저장한다.
    - 접근과 탐색이 더 쉽고 빠르다.
- Graph Based Store
    - 데이터를 나타내고 저장하기 위해 그래프 모델을 이용한다.
    - 시각화, 분석, 연관성 발견에 효과적이다.

Services for NoSQL

- Key-Value Store : Redis, Memacached, DynamoDB
- Document Store : MongoDB, DocumentDB, CouchDB
- Column Store : Cassandra, Hbase
- Graph Store : Neo4J, CosmosDB

Advantages of NoSQL

- 많은 양의 정형, 반정형, 비정형 데이터를 다룬다.
- 다수의 데이터센터에서 분산시스템 운영가능하다.
- 효율적이고 비용효율적인 확장가능한 구조를 가지고 있다.
- 확장성을 향상하고, 통제가 쉬우며, 간단한 디자인을 가지고 있다.

4.4 Data Warehouse, Data Marts, and Data Lakes

Date Warehouse

데이터 웨어하우스(Data Warehouse)는 다양한 소스에서부터 온 데이터들이 통합된 중앙 저장소이다. 데이터 웨어하우스에는 데이터들이 특정한 목적으로 정형화 되어있다. 이전에 데이터 웨어하우스는 CRM, ERP, HR, Finance Applications로부터 저장된 관계형 데이터가 존재했다. 하지만, NoSQL의 등ㅇ장으로 비 관계형 데이터 저장소 또한 데이터 웨어하우스에 사용되고 있다.

데이터 웨어하우스는 세가지 구조를 가지고 있다.

- Database Server : 다른 소스들로부터 데이터를 추출한다.
- OLAP Server : 데이터베이스 서버로부터 온 정보를 분석하고 처리한다.
- CLient Front-End Layer : 데이터를 질문, 분석, 보고한다.

요즘 데이터 웨어하우스는 클라우드 기반으로 운영된다. 클라우드 데이터 웨어하우스는 비용이 적고, 저장공간이 무한하며, 장애 복구에 속도가 빠르다.

Data Marts

데이터 마트(Data Marts)는 데이터 웨어하우스의 하위 부분으로, 특정한 비즈니스 목적, 커뮤니티에 의해 상성된다. 데이터 마트는 세가지 타입으로 나뉜다.

- Dependent : 기업 데이터 웨어하우스의 하위 부분, 제한적이고 안정성을 제공함
- Independent : 기업 외부 데이터에서 데이터를 가져올 수 있음.
- Hybrid : Dependent + Independent

데이터 마트는 유저들이 원하는 가장 적절한 데이터를 제공하여 비즈니스 작업 수행을 가속화 시킨다. 또한 비용과 시간면에서 효율적인 방식을 제공하여 유저들의 응답시간을 향상 시킬 수 있다.

Data Lakes

데이터 레이크(Data Lake)는 많은 양의 정형, 반정형, 비정형 데이터를 Raw Format으로 저장하는 데이터 저장소이다. 데이터 레이크에 저장되는 데이터들은 구조와 Schema가 정의되지 않으며 소스에서 온 Raw Data형태로 저장되어 사용에 따라 변형된다.

데이터 레이크는 기술에 독립적인 Reference 구조를 가지고 있다. 분석가와 데이터 사이언티스트들을 위한 데양한 기술을 결합하여 빠른 데이터 탐색을 가능하게 한다.

4.5 ETL, ELT, and Data Pipelines

ETL

ETL는 Extract, Transform, Load의 줄임말로, Raw Data가 Analysis-Ready data로 바뀌는 과정이다. ETL은 자동화된 공정으로 분석과 보고하고자 하는 데이터를 데이터 소스에서 추출하고, 조직에서 사용가능한 형태로 변환하여 적재하는 과정이다.

1) Extract

- 변환을 위해 필요한 데이터를 추출하는 단계
- Batch Processing을 통해 데이터를 추출(Stitch, Blendo)
- Streaming processing을 통해 실시간으로 전송되는 데이터를 추출(Apache Kafca, Apach Storm)을 이용한다.
- Batch Processing : 최종 사용자의 개입 없이 실행을 스케쥴링할 수 있는 작업의 실행. 컴퓨터 프로그램 흐름에 따라 순차적으로 자료를 처리하는 방식.

2) Transform

- 데이터를 분석가능하도록 규칙과 함수를 사용해서 변환하는 작업
- 데이터 형식과 측정 단위를 표준화한다.
- 중복 데이터를 제거한다.
- 필요하지 않은 데이터를 제거하고 파싱한다.
- 데이터 테이블 간 Key 관계를 설립한다.

3) Load

- 변형된 데이터를 데이터 저장소로 적재하는 단계
- Initial Loading : 모든 데이터를 저장소에 이동시킴
- Incremental Loading : 업데이트 되거나 변경된 데이터를 주기적으로 적용
- Full Regresh : 데이터 테이블을 지우고 새로운 데이터를 적재함

이전에는 큰 규모의 데이터를 batch시키는 방법으로 데이터를 적재하였다면, ETL streaming tool들이 등장으로 실시간 데이터를 적재하는 방법이 유용하게 사용된다.

ELT

ELT(Extract-Load-Transform) 방식은 Cloud 기술이 등장하면서 적용되는 방식이다.

- 비정형, 비 관계형 데이터를 처리하는데 유용하다.
- 데이터 레이크에 이상적이다.
- 추출-적재에 걸리는 시간이 줄어든다.
- Raw Data를 즉시 사용가능하다.
- EDA에 유연성을 제공한다.
- BigData 처리방식에 더 적합하다.

Pipelines

Data Pipeline은 데이터가 추출부터 시스템에 적용되는 모든 단계를 동봉하는 과정으로 ETL, ELT가 Data Pipeline에 포함되어 있다. 데이터 파이프라인은 실시간으로 전송되고 처리되는 데이터 흐름에 유용하고, Long-running batch query와 Small interactive query를 동시에 지원 가능하다.

4.6 Data Integration Platforms

데이터 통합(Data Integration)은 서로 다른 소스에 있는 데이터를 결합시키고 사용자에게 통합된 보기를 제공하는 과정이다. 앞서 사용되었던 ETL, ELT가 데이터 통합 과정의 일부분으로 작동하게 된다. 만들어진 커넥터와 어댑터들은 다양한 소스에서 제공된 데이터의 흐름을 연결하고 구축할 수 있도록 도와준다. 또한 Batch Processing, Streaming Processing을 최적화 해주고, Data Governance, Compliances, Security 를 위한 부가적인 기능을 제공한다.

5. Big Data Platforms

5.1 Foundations of Big Data

빅데이터(Big Data)는 사람, 기계, 툴에 의해 만들어지는 많고 다양한 부피의 데이터이다. 빅데이터는 실시간으로 비즈니스 인사이트를 위해 모이는 많은 양의 데이터를 모으고 처리하고, 분석하는 새롭고, 혁신적이고, 확장성 있는 기술을 요구한다.

기존 대이터베이스 관리도구의 능력을 넘어서는 대량의 정형 또는 비정형 데이터의 가치를 추출하고 결과를 분석하는 기술이다.

5.2 BIg Data Processing Tools

빅데이터 처리 기술은 정형, 비정형, 반정형 데이터를 다루는 방법을 제공함으로써 데이터에서 가치를 추출할 수 있게 해준다.

Hadoop

하둡(Hadoop)은 빅데이터를 처리하고 분산형 저장기술을 제공하는 Tool들의 집합이다. 아파치 하둡(Apache Hadoop)은 대량의 자료를 처리할 수 있는 큰 컴퓨터 클러스터에서 동작을 지원하는 분산 응용 프로그램을 지원하는 프리웨어 자바 소프트웨어 프레임워크이다. 분산 처리 시스템인 하둡 분산 파일 시스템과 맵리듀스를 구현한 것이다.

HDFS

HDFS는 Hadoop Distributed File System의 약자로 수십 테라바이트 또는 페타바이트 이상의 대용량 파일을 분산된 서버에 저장하고, 그 저장된 데이터를 빠르게 처리할 수 있게 하는 파일 시스템이다. 또한 저사양의 서버를 이용해서 스토리지를 구성할 수 있어 기존의 파일 시스템에 비해 장점을 가진다.

Hive

아파치 하이브(Apache Hive)는 하둡에서 동작하는 데이터웨어하우스 인프라 구조로써 데이터 요약, 질의 및 분석 기능을 제공한다. 아파치 하이브는 HDFS나 Hbase와 같은 데이터 저장 시스템에 저장되어 있는 대용량 데이터 집합들을 분석한다 HiveQL이라고 불리는 Query Language를 사용하여 맵 리듀스의 모든 기능을 제공한다.

Spark

아파치 스파크는 오픈 소스 클러스터 컴퓨팅 프레임워크이다. 인메모리 기반의 대용량 데이터 고속처리를 지원하여 빠른 처리를 보장하고, 다양한 언어를 지원한다.