CM2606 Data Engineering

Big Data Processing 01

Week 07 | Piumi Nanayakkara

Learning Outcomes

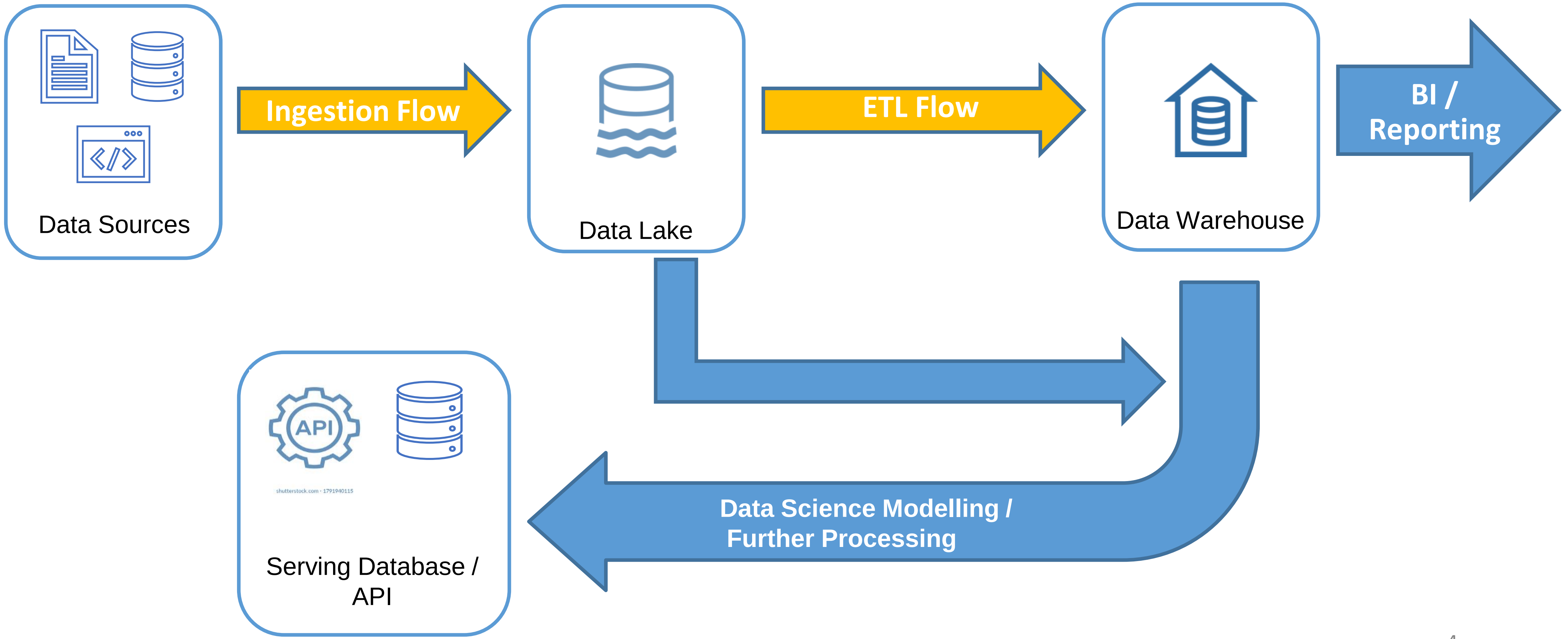
- Covers LO2 and LO4 for Module

- On completion of this lecture, students are expected to be able to:
  - Apply knowledge of access connectors, Ingestion mechanisms, transformation rules and data loading mechanisms to produce stable data pipeline designs

Content

- Data Ingestion
    - CDC – Change Data Capture
    - Access Connectors
    - Batch Vs Streaming
- Data Profiling
- Data Transformations
    - Types, Benefits, Challenges
- Data Loading

Data Pipeline: Common Usage



Data Ingestion

- Data Ingestion refer to the process of extracting row data from source data systems into the organizations data eco system.

- Early days before big data came into visibility, this was same as the extraction step of ETL pipelines where data was extracted, transformed and loaded into the warehouse in a single run.
    - a simple microservices workflow would have covered all three stages

Data Ingestion

- Need for Separation:
  - When dealing with big data it is recommended to have row data staged before applying any transformations on top of it.
  - This enables to keep a reference in handling any failures in the later parts of the pipeline and smooth back tracking.
  - This manual check point is important since most of the time big data is processed using distributed processing where auto rollback would be complex and costly to handle.

Data Ingestion Initiation – Push Vs Pull

- **Pull based Approach:**
  - Data ingestion is initiated by the receiving end from the source like a database, file system, a queue or an API
  - Target system polls at different frequencies and checks if new changes are available in source

- **Push based Approach:**
  - Data ingestion is initiated by the data sources by pushing / publishing new or changed data at different frequencies.
  - Leads to lower latency between the source and target
  - Applications can directly push data into your data lake, but it is always recommended to have a messaging platform as Kafka in between, so target can read in its own pace.

Change Data Capture (CDC)

- CDC is the process of recognizing when data has been changed in a source system so a downstream process or system can action that change.

- Capturing the Change as soon as it occurs helps an organization to stay in sync with source data

- Applicable to both Pull Based and Push Based approaches.

CDC Methodologies

- Row Versioning
  - Target system uses reference tables that for each ID, stores the last known version.
  - The target then checks if any rows have a version number greater than that stored in the reference table.
  - If they do, then these records are captured, and the changes reflected in the target system.
  - The reference table then also needs updating to reflect the new version number for these records.

Customer ID	Email Address	Version
123	jan.perera@gmail.com	0

Customer ID	Email Address	Version
123	j.perera@gmail.com	1

CDC Methodologies

- Update Timestamps

  - Every time a record in the database changes, you update a column. Instead of this column storing the version number of the record, it stores a timestamp of when the record was changed.

  - Drawback: We no longer know how many times the record has been changed

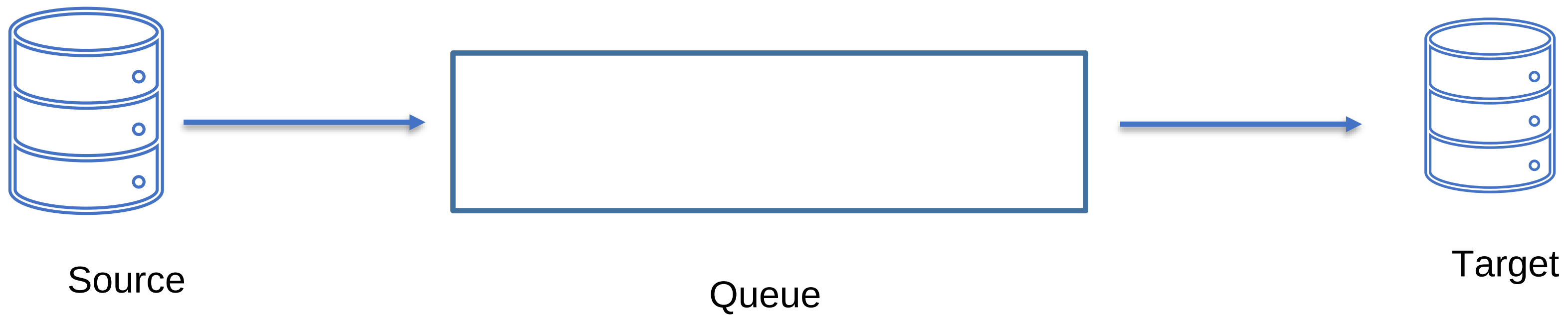
  - The target system can now just request any records from the source system that have an update timestamp greater than the latest one they have in their system.

Customer ID	Email Address	Update_timestamp
123	jan.perera@gmail.com	2022-01-01 00:00:00

Customer ID	Email Address	Update_timestamp
123	j.perera@gmail.com	2022-01-31 00:00:00

CDC Methodologies

- Publish and Subscribe Queues
    - This pattern uses a push rather than pull approach
    - Any time a change is made to the data in the source system, the source pushes the change to the queue.
    - The target system is listening and can then consume the changes as they arrive
    - Provides high scalability when source systems produces many updates in quick time
    - The two systems will be decoupled and allowed to change database structure on its own.



Data Ingestion – Access Connectors

- **Database Connectors:**
  - Database connectors can be used for importing data from relational database management systems into big data storage and analytics frameworks
  - E.g., JDBC Connectors

- **Custom Connectors:**
  - Custom connectors can be built based on the source of the data and the data collection requirements.
  - E.g., custom connectors for social networks based on REST, WebSocket
    custom connectors for ERP applications such as SAP / Oracle

Data Ingestion – Access Connectors

- **Publish-Subscribe Messaging:**
  - Publish-Subscribe is a communication model that involves publishers, brokers and consumers.
  - Publishers are the source of data. Publishers send the data to the topics which are managed by the broker.
  - E.g., Apache Kafka

- **Messaging Queues:**
  - Messaging queues are useful for push-pull messaging where the producers push data to the queues and the consumers pull the data from the queues.
  - The producers and consumers do not need to be aware of each other.
  - E.g., RabbitMQ, ZeroMQ, RestMQ and Amazon SQS.

Data Ingestion – Access Connectors

- **Source-Sink Connectors:**
  - Source-Sink connectors allow efficiently collecting, aggregating and moving data
    - from various sources: server logs, databases, social media, streaming sensor data from Internet of Things devices and other sources
    - into a centralized data store
  - E.g., Apache Flume, which is a framework for aggregating data from different sources. Apache Flume is used to collect log data present in log files from web servers and aggregating it into HDFS for analysis.

Data Ingestion – Batch Ingestion

- Data is ingested as an accumulated group, often at a defined time and frequency
- Data will be loaded into staging area after converting into a common file format.
- Appropriate partitioning strategy needs to be followed
- E.g., Daily records are ingested at midnight from an operational database
- Characterized by data volume

Data Ingestion – Streaming Ingestion

- Data is ingested as they produced in real time
- Depending on the requirements of the organization data will be either stored in staging area to be processed as batches or will be fed to real time processing engines.
- E.g., Data could come in via Kafka, Kinesis
- Characterized by data arrival rate

Data Profiling

- Before transforming data ingested from source systems, analysis should be carried out understand and identify the data in the original format.

- A critical component of implementing a data strategy

- Helps you discover, understand and organize your data.
    - Verifying that the information in your tables matches the descriptions
    - Revealing the relationships that span different databases, source applications or tables.

Data Profiling - Categories

- Structure Discovery:
  - Validates that the data that you have is consistent and formatted correctly
    - E.g., use of pattern matching for mobile numbers
  - Examines simple basic statistics in the data like the minimum and maximum values, means, medians, modes and standard deviations
- Content Discovery
  - Find areas that contain null values or values that are incorrect or ambiguous
- Relationship Discovery
  - Understanding of the connections between the data sets

Data Profiling - Techniques

- Column profiling:
  - Scans through a table and counts the number of times each value shows up within each column.
  - This method can be useful to find frequency distribution and patterns within a column of data.

- Cross-column profiling:
  - Made up of two techniques help analyze dependencies among data attributes within the same table.
  - Key analysis: examines collections of attribute values by scouting for a possible primary key.
  - Dependency analysis: determines whether there are relationships or structures embedded in a data set.

Data Profiling - Techniques

- Cross-table profiling:
  - Foreign key analysis, examine the relationships of column sets in different tables.
  - This can help cut down on redundancy but also identify data value sets that could be mapped together.

- Data rule validation:
  - Verify that data instances and data sets conform with predefined rules.
  - This process helps find ways to improve data quality and can be achieved either through batch validation or an ongoing validation service.

Data Transformations

- Refers to change of format, structure or values of data
- Could be performed either before loading data into data warehouse (ETL) or after loading data into row/staging tables in data warehouse (ELT)

Data Transformation Types

- Constructive Transformation:
  - Adding, Copying, and Replicating data
- Destructive
  - Deleting fields and records
- Aesthetic
  - Standardizing: salutations or street names
- Structural:
  - Renaming, Combining columns

Data Transformations: Formatting Data

- Type Conversions / Schema Mappings
  - Date Formatting
  - Text to Int Conversions, e.g., Quantities or Totals
  - Int to Text Conversions e.g., Customer IDs
  - Column Renaming

- Data Parsing
  - Text data to relational Structures, E.g., Comma de limited logs
  - Flatten Hierarchies, E.g., Web scraped data in json/xml formats
  - Split long or freeform fields into multiple columns

Data Transformations: Cleaning Data

- Data Filtering
  - Filtering out unnecessary fields, columns, and records
    - E.g., records from business regions that aren’t of interest
- Ambiguity Handling
  - For categorical variables same value represented by two strings
    - E.g., In T shirt sizes “S” and “Small” both being used
- Duplication Handling
- Corrupt Data Handling

Data Transformations: Missing Value Handling

- Drop the Rows
  - Check missing value ratios

- Replace the Values
  - Mean / Median / Mode
  - Zero
  - Interpolate / Extrapolate

Data Transformations: Scaling

- Min-Max Scaler:
  - Shrinks the feature values between any range of choice. For example, between 0 and 5.
- Normalization
  - Process of scaling in respect to the entire data range so that the data has a range from 0 to 1.
- Standardization
  - Process of transforming in respect to the entire data range so that the data has a mean of 0 and a standard deviation of 1. It’s distribution is now a Standard Normal Distribution.
- Transformation
  - Application of the same calculation to every point of the data separately.
  - Normalization and Standardization can be seen as special cases of Transformation

Data Transformations: Enrichment

- Aggregation
  - Transforming a series of customer transactions to hourly or daily basis.
- Denormalized Schemas
  - E.g., Add monthly expenditure of a customer in the customer table
- Merging
  - Data from different sources can be merged

Data Transformations: Sanitization

- Anonymization
  - Removing personally identifiable information from data sets

- Encryption
  - Can perform encryption at multiple levels, from individual database cells to entire records or fields.

Data Transformations: Sanitization

Personal Identifier	Quasi Identifiers			Sensitive Attribute
Name	**Age**	**Gender**	**Zip Code**	**Disease**
Joseph	28	M	10230	Heart Disease
Gamage	34	F	67432	Cancer
Fernando	55	M	10765	Cancer
Rashid	44	M	23000	Flu

Data Transformations: Sanitization

Personal Identifier	Quasi Identifiers			Sensitive Attribute
Name	**Age**	**Gender**	**Zip Code**	**Disease**
Joseph	20 -29	Any	10***	Heart Disease
Gamage	30 - 39	Any	67***	Cancer
Fernando	50 - 59	Any	10***	Cancer
Rashid	40 - 49	Any	23***	Flu
	Generalized	Generalized	Suppressed	

Benefits

- Better-organized Data
  - Easier for both humans and computers to use

- Getting maximum value out of data

- Improves data quality
  - Protects applications from potential landmines such as null values, unexpected duplicates, incorrect indexing, and incompatible formats.

- Facilitates compatibility between applications, systems, and types of data
  - Data used for multiple purposes may need to be transformed in different ways.

Challenges

- Costs associated
  - Specific infrastructure, software, and tools
  - Licensing of software
  - Talent acquisition.
  - Resource Intensive

- Loss of valuable information
  - Lack of expertise and appropriate subject matter expertise
  - Ranges extended, values being replaced
  - Irreversible actions could be performed

Data Loading

- Refers to the end of the ETL pipelines where data is loaded into data warehouses.

- Data pipelines would be designed to do required transformations and validations before / after loading data such as
  - Surrogate Key Assignments
  - Foreign key Constraint checks
  - Indexing

- At the end of the data load any triggers to reorganize data inside the data warehouse or any dashboard data refresh would be followed.

- One of the critical steps because, If there is a delay in loading the data, then the data warehouse user applications will be impacted. Therefore, it is very important to tune the data load first.

Data Load Tuning

- Approach 1:
    - The very common approach is to insert data using the **SQL Layer**.
    - Normal checks and constraints need to be performed. When the data is inserted into the table, the code will run to check for enough space to insert the data. If sufficient space is not available, then more space may have to be allocated to these tables.
    - These checks take time to perform and are costly to CPU.

- Approach 2:
    - Bypass all these checks and constraints and place the data directly into the preformatted blocks.
    - These blocks are later written to the database.
    - It is faster than the first approach, but it can work only with whole blocks of data. This can lead to some space wastage.

READING

- Apache Hive Warehouse Connector Use-Cases - Cloudera Blog