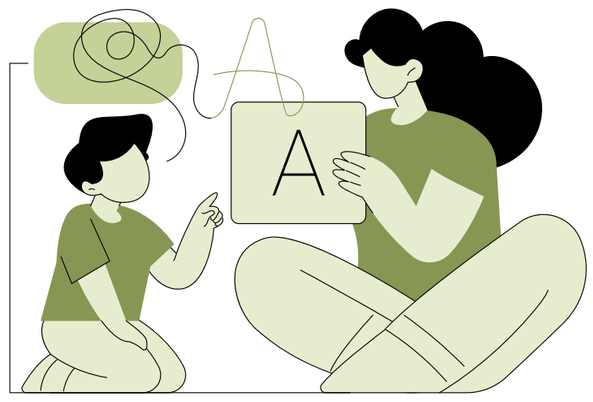
Mastering

Incremental Data Loading w

Azure Data Factory

What is Incremental Data Loading? 🤔



In the fast-paced world of data management, Incremental Data Loading offers a smart and efficient way to handle data. Here's an in-depth look at what it is, why it's valuable, and how Azure Data Factory (ADF) leverages this approach.

Incremental Data Loading, aka delta loading, focuses on updating only the parts of your data that have changed since the last load. This contrasts with full load operations, which involve refreshing the entire dataset every time.

Why is Incremental Loading Essential? 🌟

1

Boosted Efficiency 🚀

By only processing new or altered data, this method minimizes the volume of data handled, leading to significant savings in time and computational resources.

2

Timely Updates ⏱️

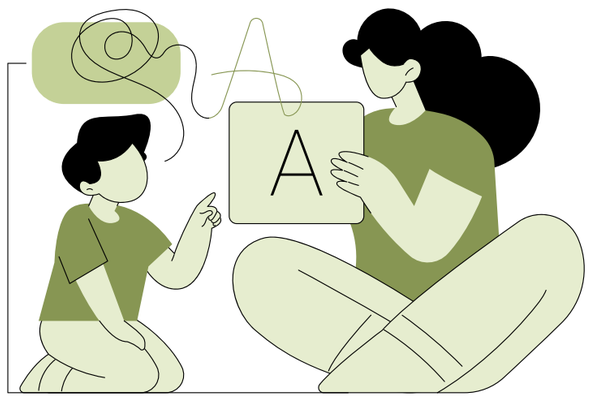
It enables more frequent and relevant updates without the overhead of reloading the entire dataset.

3

Reduced Network Load

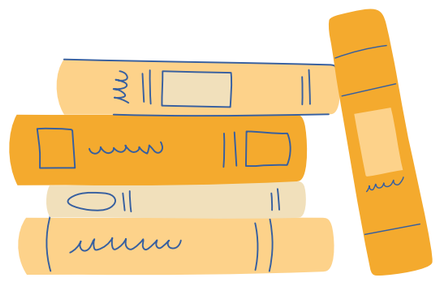
Smaller data transfers alleviate the strain on network resources, enhancing overall network performance.

How Azure Data Factory Facilitates Incremental Loading 🔧



1 Source Data Store 🏬

This can be an on-premises SQL Server, Azure SQL Database, or any other supported source.

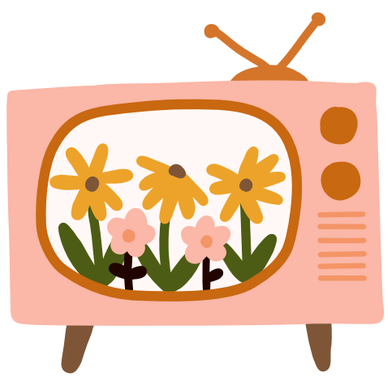


2 Sink Data Store 📥

The destination for the data, such as Azure SQL Database or Azure Data Lake.

3 Watermark 🏷️

A key component in this process, the watermark column tracks the last modification or creation time of records



4 Integration Runtime (IR) ⚙️

ADF uses Integration Runtimes for data movement. For on-premises sources, a Self-Hosted IR is required.

5 Linked Services 🔗

Define how ADF connects to both source and destination data stores.

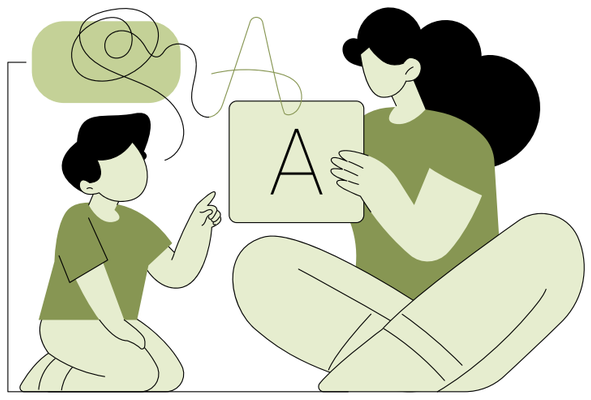
6 Datasets 📊

Represent the data structures that ADF interacts with within the data stores.

Pipeline 🛠️

Manages the entire data loading process, including lookups for watermark values, data copying, and watermark updates.

Steps for Incremental Data Loading 🔁



1

Get Last Watermark Value ⏮️

Retrieve the timestamp or identifier of the most recent processed data.

2

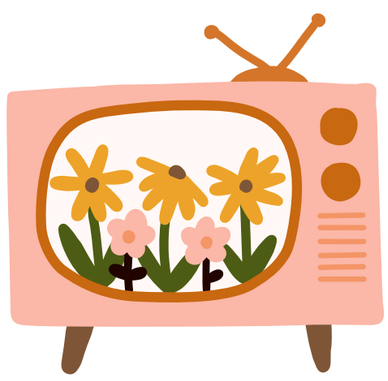
Obtain New Watermark Value ⏭️

Fetch the latest watermark from the source.

3

Extract Data Between Watermarks 📋

A key component in this process, the watermark column tracks the last modification or creation time of records



4

Load Data into Staging Table 📥

Place the extracted data into a staging area in the destination.

5

Merge Staging Data 🔄

Integrate the staging data into the final table in the destination store.

6

Update Watermark 🆙:

Save the new watermark value to prepare for the next load.