BITS Pilani presentation

BITS Pilani
Pilani Campus

Dr. Vivek V. Jog
Dept. Of Computer Engineering

innovate achieve lead

BITS Pilani

Pilani Campus

Big Data Systems (S1-24_CCZG522)

Lecture No.5

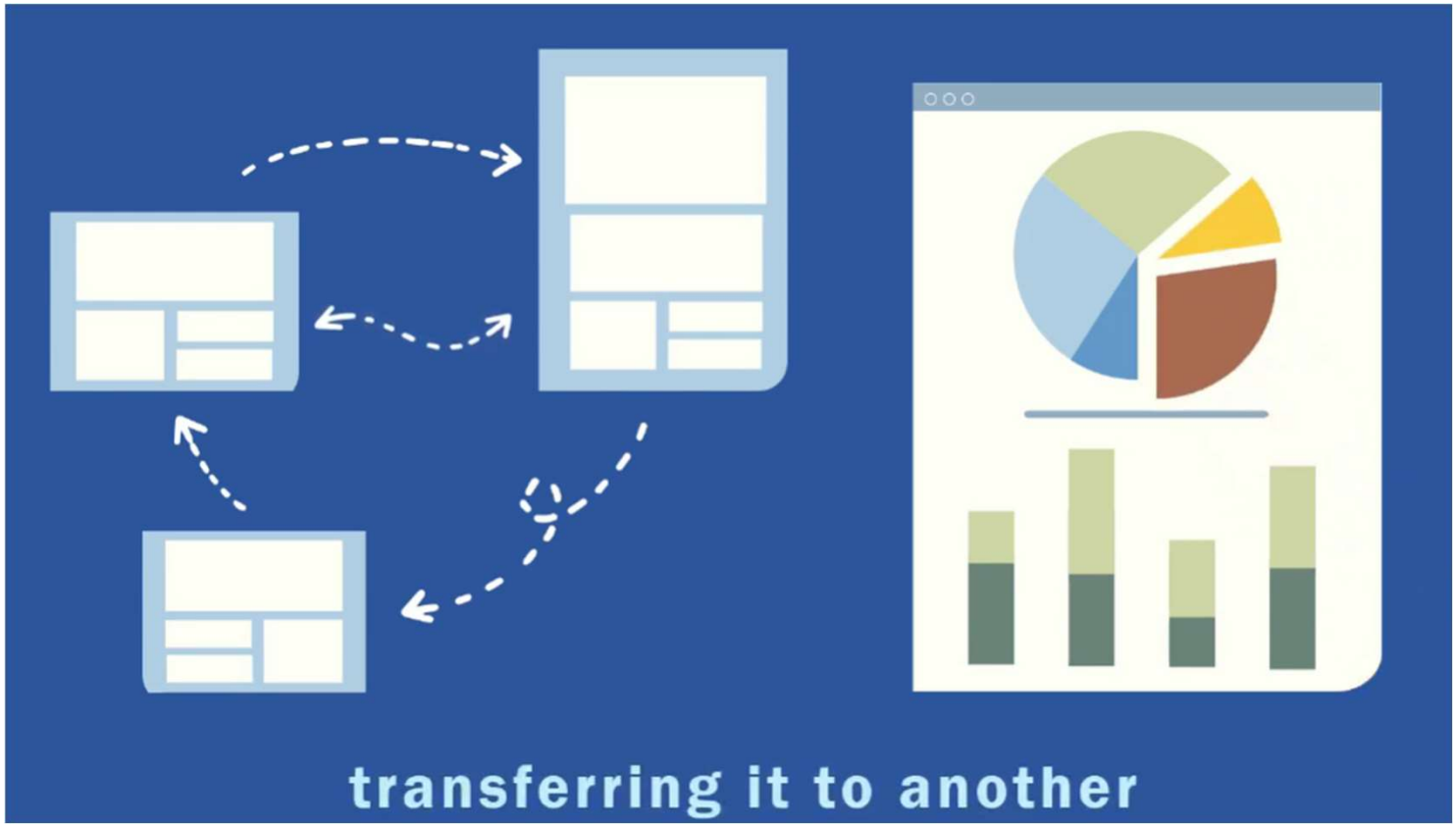
Data Extraction

What is Data Extraction?

Extracting data is key in managing and analyzing information. As firms collect stacks of data from different places, finding important info becomes crucial. We gather specific info from different places like databases, files, websites, or APIs to analyze and process it better. Doing this helps us make smart decisions and understand things better.

Gathering data from various places, changing it so we can use it, and putting it where we need it for review is what data extraction is about.

Data Extraction – Capture and Transform



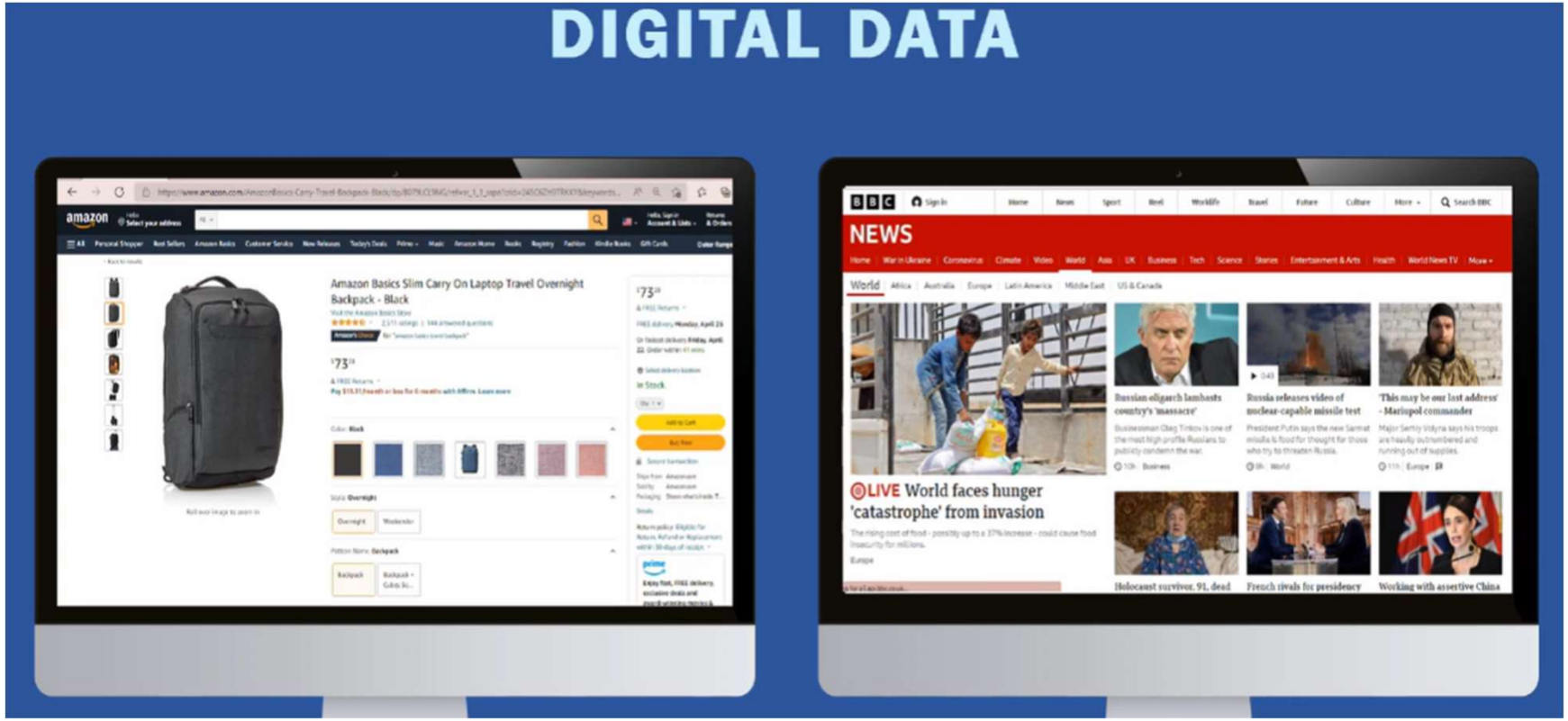
Why we need Data Extraction ?

1. **Facilitating Decision-Making:** It gives us what has happened (historical trends), what's happening (current patterns), and what might happen (emerging behaviours). This helps firms or organizations make plans for better business.
2. **Empowering Business Intelligence:** Business smarts need relevant and timely data for helpful insights. This makes a group more focused on data.
3. **Enabling Data Integration:** Firms often hold data in different systems. Taking out the data makes it mix better. This gives an all-around and fitting view of firm-wide data.
4. **Automation for Efficiency:** Automated data extraction processes boost efficiency and less hands-on need. Automation offers a smooth, steady way to deal with lots of data.

Data Extraction Process

- **Filtering:** It demands meticulous sifting through information, delicately pinpointing and extracting details that harmonize with predetermined standards. This pivotal stage ensures that data emerges, filtering noises and elevating the precision of subsequent evaluations.
- **Parsing:** Is a systematic exploration of the data framework, unravelling it into elements that can be further maneuvered and processed. This holds particular importance when grappling with information that lacks a clear structure or adheres to a semi-organized format.
- **Structuring:** Raw data often lacks a coherent arrangement. In the journey of data extraction, an art known as organizing is employed to structure and format the information in a manner conducive to analysis process.

Extract Product information from an E-Commerce Website



Data Extraction from CSV using Python


```python
import pandas as pd
from io import StringIO

# Sample in-line CSV data for example purpose
csv_data = """Name,Age,Occupation
John,25,Engineer
Jane,30,Teacher
Bob,22,Student
Alice,35,Doctor"""

# Read the CSV data into a DataFrame
df_csv = pd.read_csv(StringIO(csv_data))

# Extract data based on the 'Occupation' column
engineers_data_csv = df_csv[df_csv['Occupation'] == 'Engineer'][['Name', 'Age']]

# Display the extracted data
print("Data from CSV:")
print(engineers_data_csv)
```

Output:

```
Data from CSV:
   Name  Age
0  John   25
```

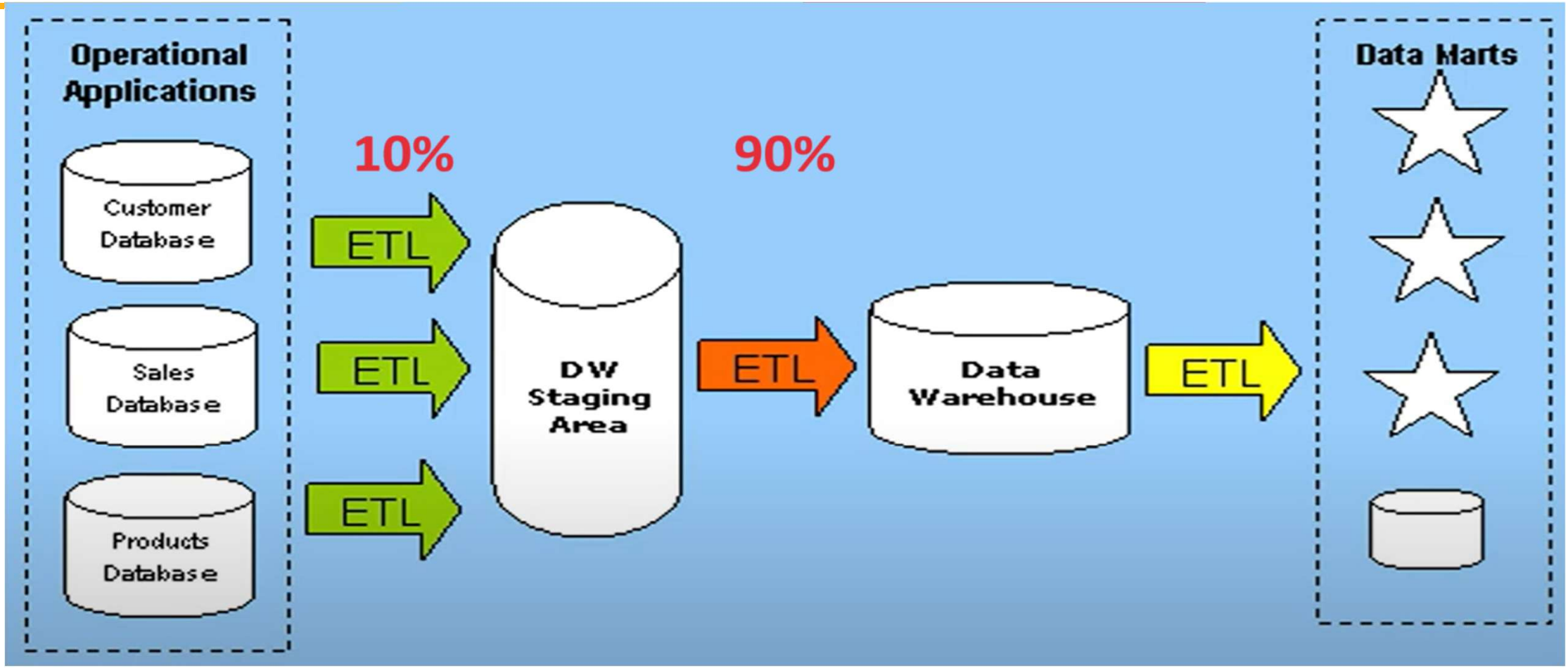
Data Extraction –Web Scraping


```python
import requests
from bs4 import BeautifulSoup

# URL for web scraping
url = "https://www.geeksforgeeks.org/basic/"

# Send a GET request to the URL
response = requests.get(url)

# Check if the request was successful (status code 20
if response.status_code == 200:
    # Parse the HTML content
    soup = BeautifulSoup(response.text, 'html.parser')

    # Extract article titles
    article_titles = [title.text.strip() for title in soup.find_all('div', class_='head')]  # Adjust the class base

    # Display the extracted data
    print("Article Titles from GeeksforGeeks:")
    for title in article_titles:
        print("- " + title)
else:
    print("Failed to retrieve the webpage. Status code:", response.status_code)
```

Output:

```
Article Titles from GeeksforGeeks:
- Array Data Structure
- Tribal Leader Vishnu Deo Sai (Biography): New Chhattisgarh Chief Minister
- Protection Against False Allegations and Its Types
- Double Angle Formulas
- MariaDB CHECK Constraint
- Cristiano Ronaldo Net Worth 2024: Football Success and Endorsements
- PM Modi Proposes to host COP 33 Summit in 2028 in India
- Why Ghol fish Declared as State Fish of Gujarat?
```

Data Transformation



Data transformation process

Data Joining

Data Duplication

Keys Restructuring

Data Cleansing

Data Validation

Data Format Revision

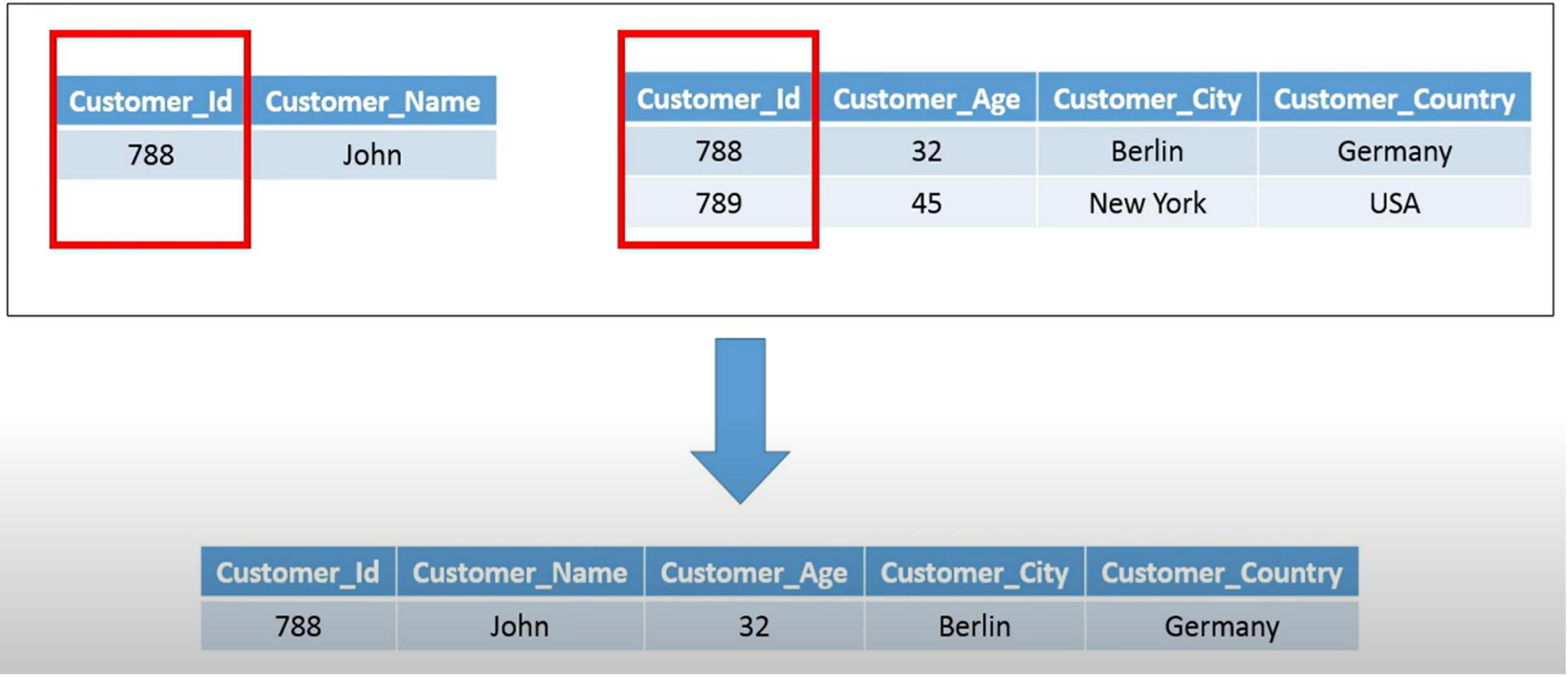
Data Derivation

Data Integration

Data Filtering

Data Splitting

Joining (two table join)

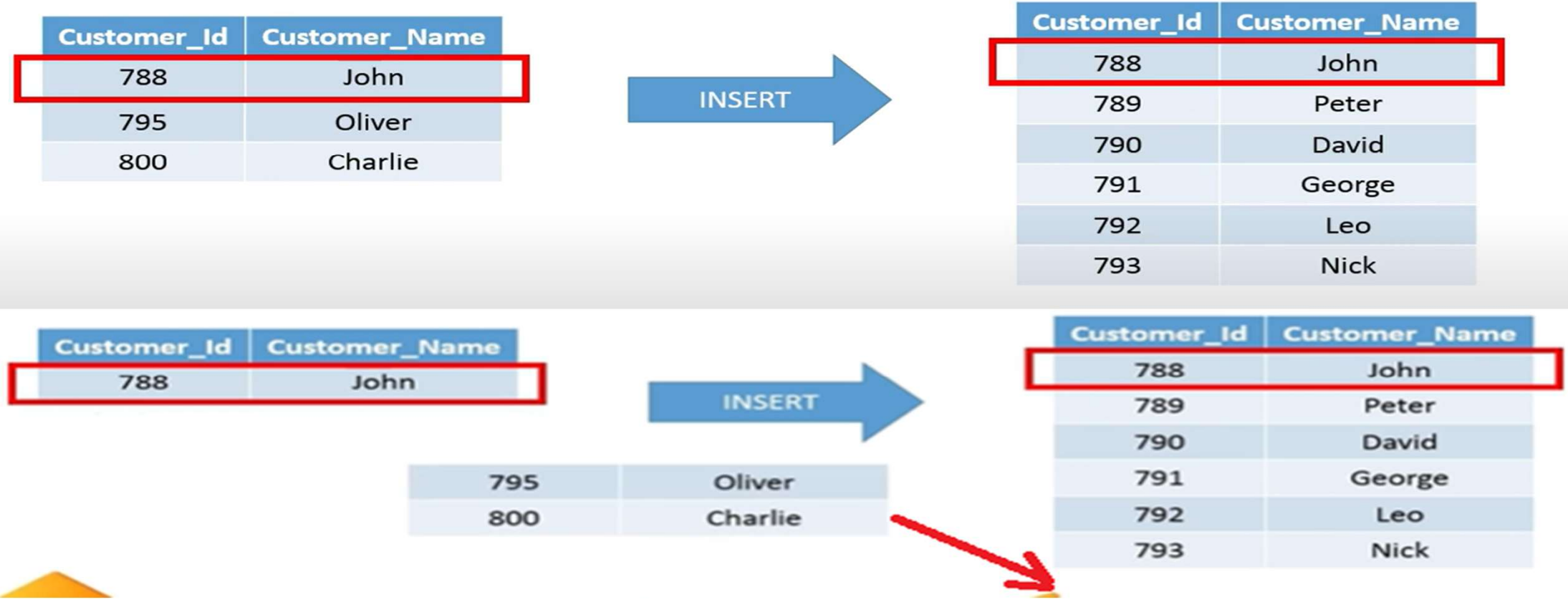


Customer_Id	Customer_Name
788	John

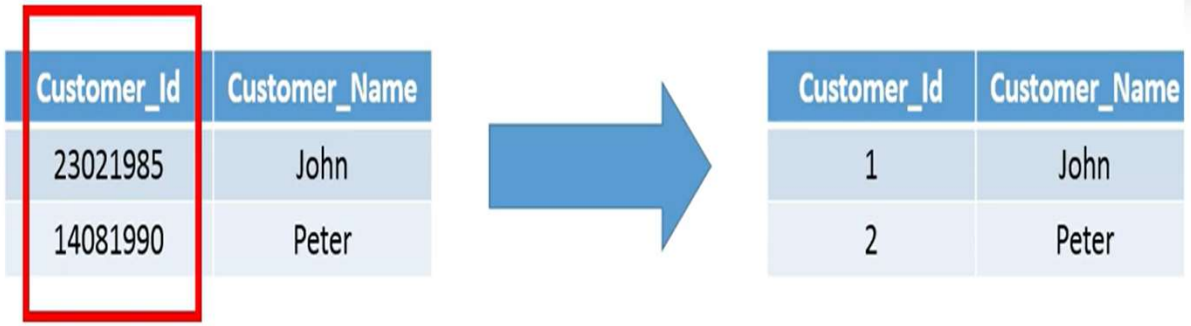
Customer_Id	Customer_Age	Customer_City	Customer_Country
788	32	Berlin	Germany
789	45	New York	USA

Customer_Id	Customer_Name	Customer_Age	Customer_City	Customer_Country
788	John	32	Berlin	Germany

Duplication



Keys Restructuring



Primary Key should be meaningless

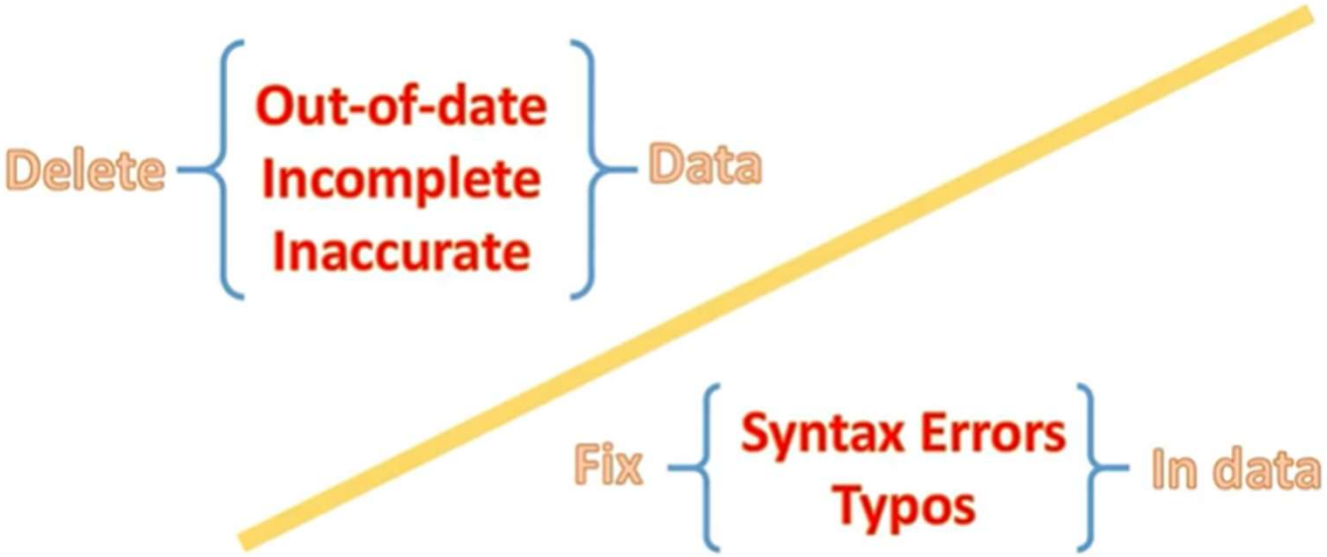
Keys: Natural vs. Surrogate

- The "Customer" example keys we just identified would be classified as *natural keys*.
- Natural keys are based on business rules and logic that determine how an individual instance can be uniquely identified.
- As we've seen, natural keys can become unwieldy, requiring a number of attributes, which makes queries difficult.
- Also, extreme care is needed as components of natural keys could change

— **Surrogate keys** are often used instead, which are system-generated unique identifiers. e.g. Customer ID, Product ID, etc.

— While surrogate keys are more efficient, important business rules are lost when they are used. ***It's a balancing act.***

Cleansing

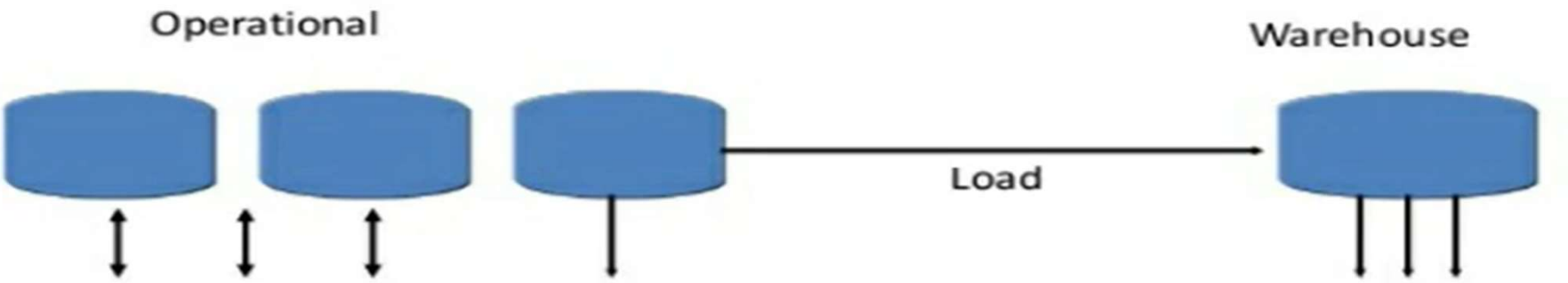

Delete
Out-of-date
Incomplete
Inaccurate
Data
Fix
Syntax Errors
Typos
In data

Integrity

Only Data Insert and Read Operations are possible on warehouse

Nonvolatile

Typically data in the data warehouse is not updated or delelted.



Validation

Range checks

- i.e. is between 1 and 100

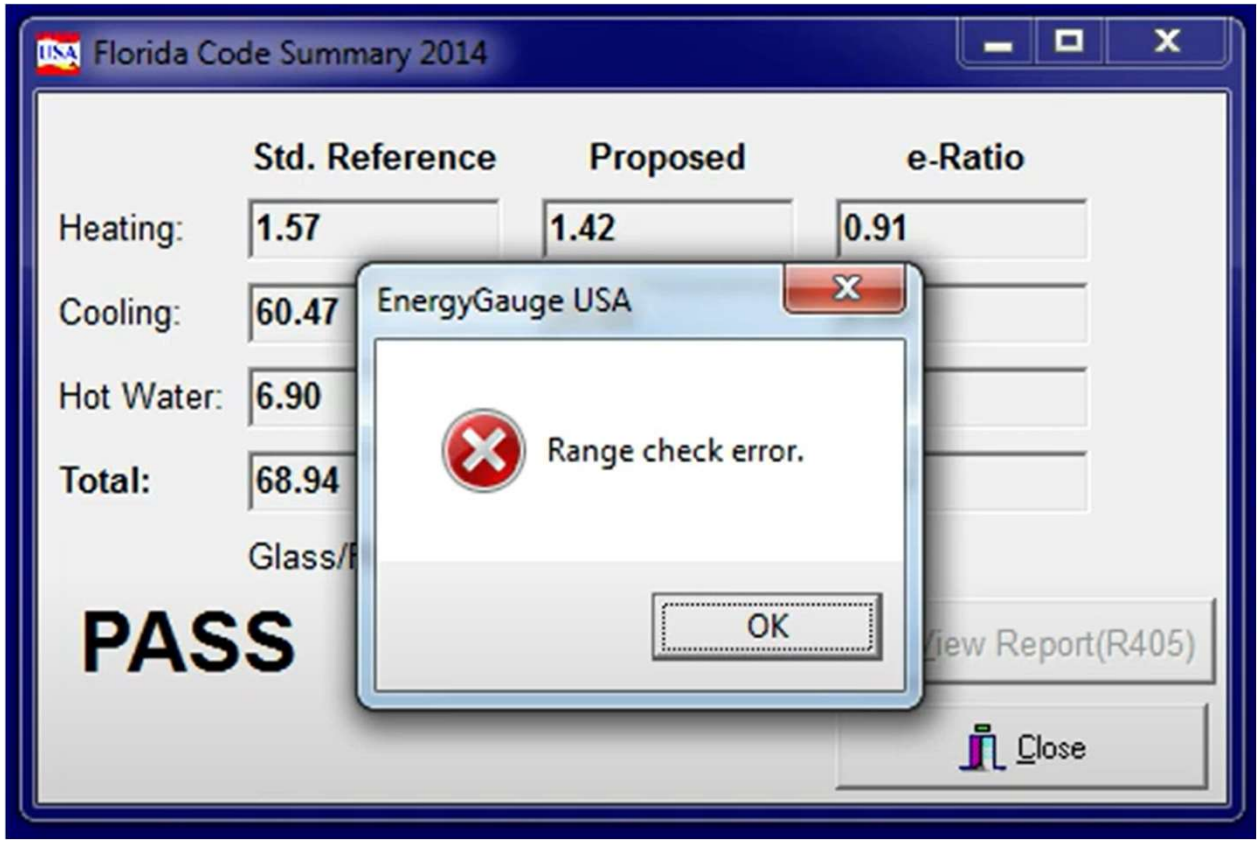
Please type in your age:

Type checks

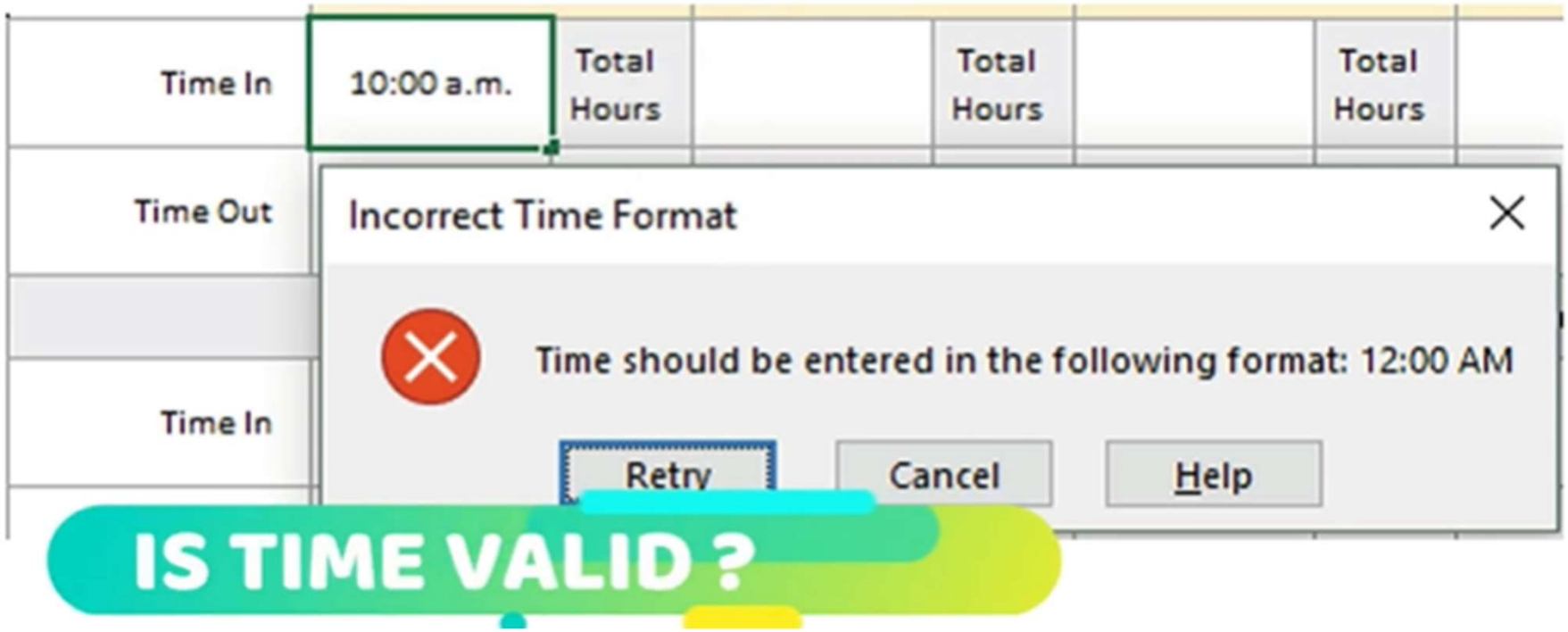
- i.e is numeric and not text

Please type in your age:

In OLTP Systems



Time Validation



Formatting

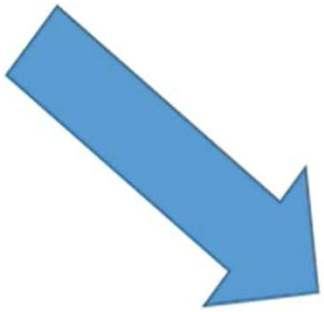
male	M
female	F

20/03/1989	20-March-1989
21/08/1999	21-April-1999

Gender	DOB
Male	1989-03-20
Female	1999-08-21

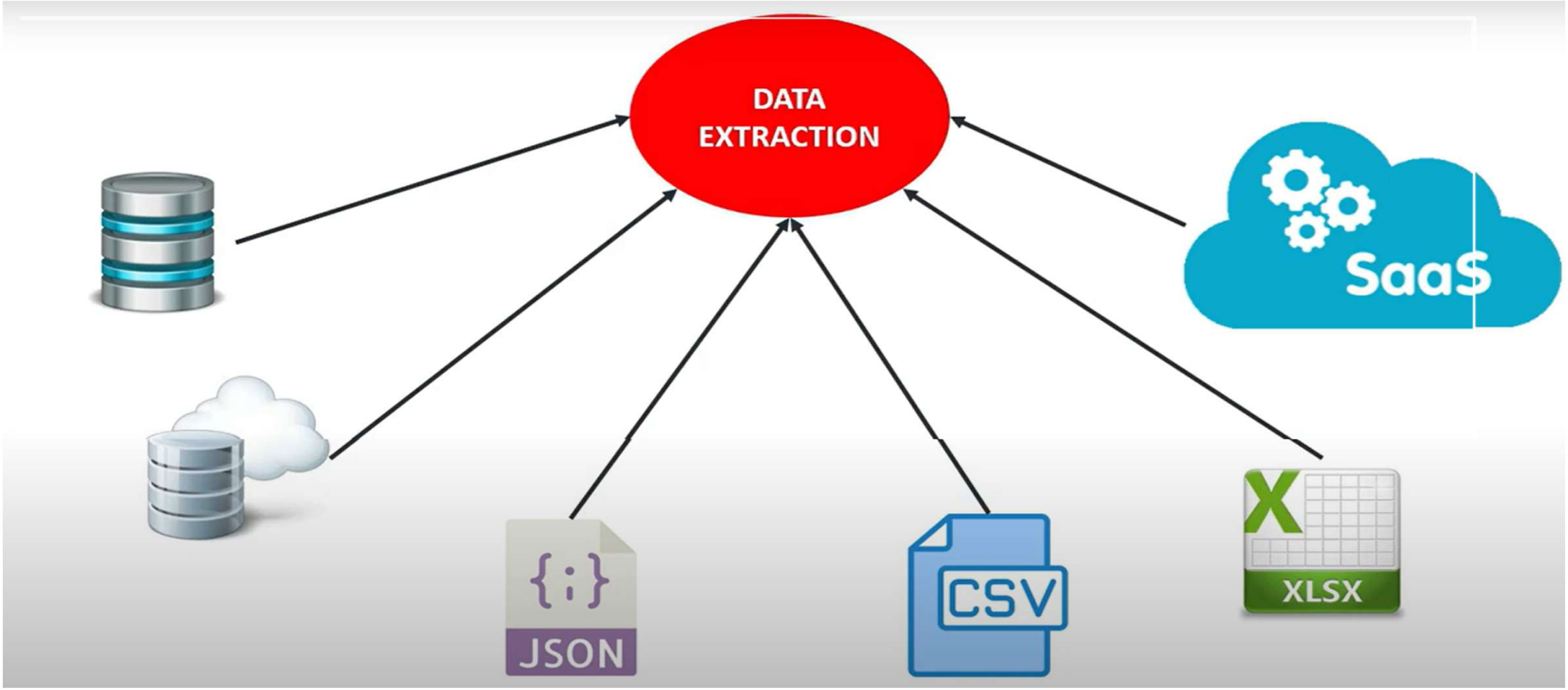
Derivation

Customer_Name	Customer_dob
John	20-08-1990
Peter	15-07-1999



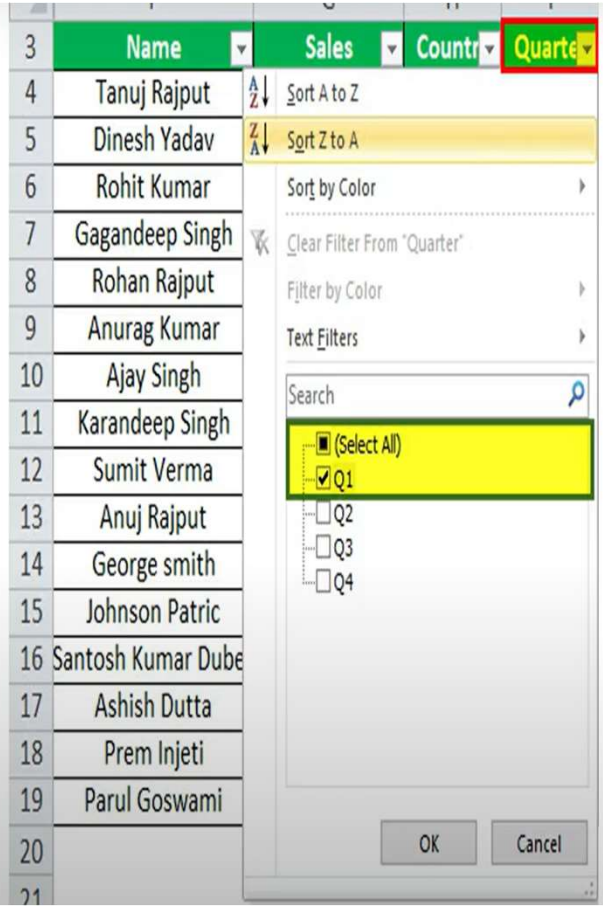
Customer_Name	Customer_dob	Customer_Age (years)
John	20-08-1990	30
Peter	15-07-1999	21

Intigration



Filter by:

- Rows
- Columns
- Unique Values



Filter by:

- Rows
- Columns
- Unique Values

Splitting

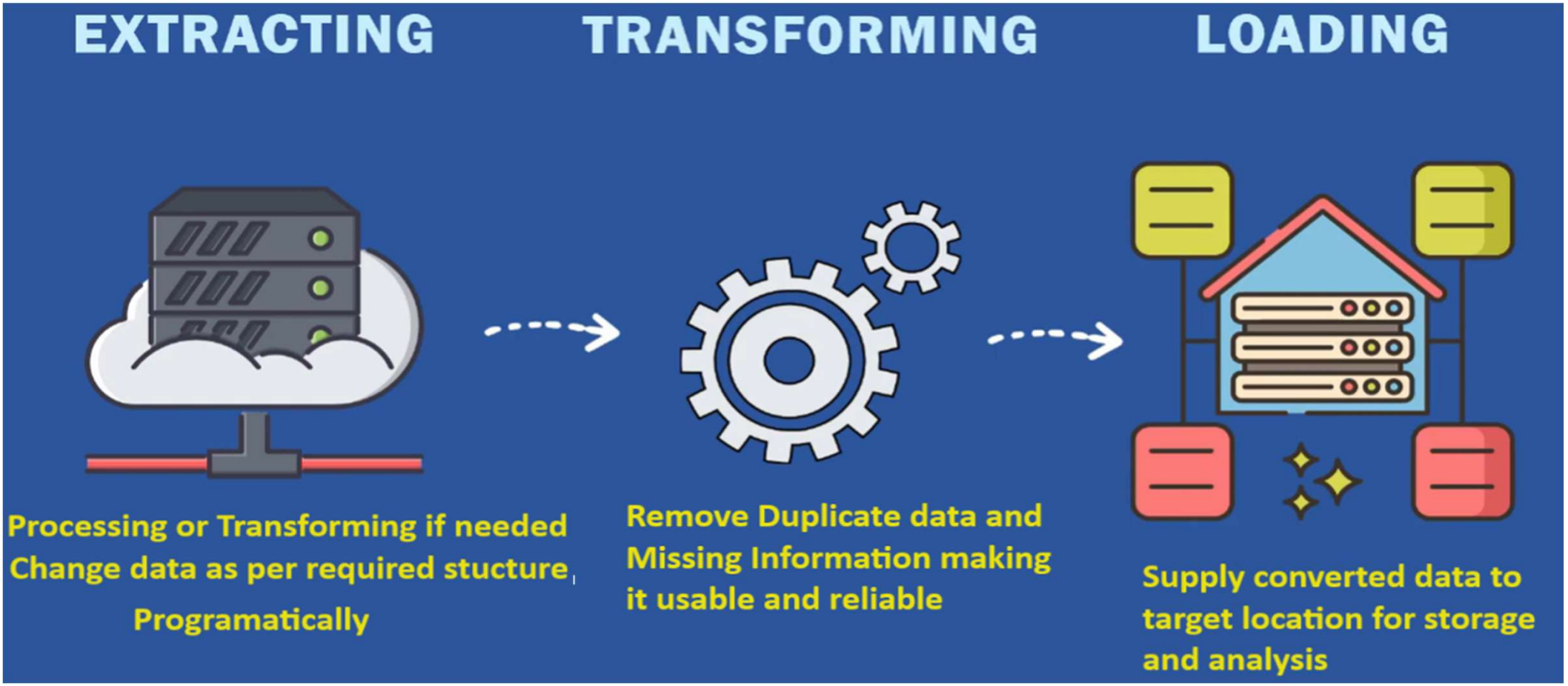
Address
Carlton Street 15, 78845, Munich; Bavaria; Germany

Street_Name	House_Number	Postal_Code	City	State	Country
Carlton Street	15	78845	Munich	Bavaria	Germany

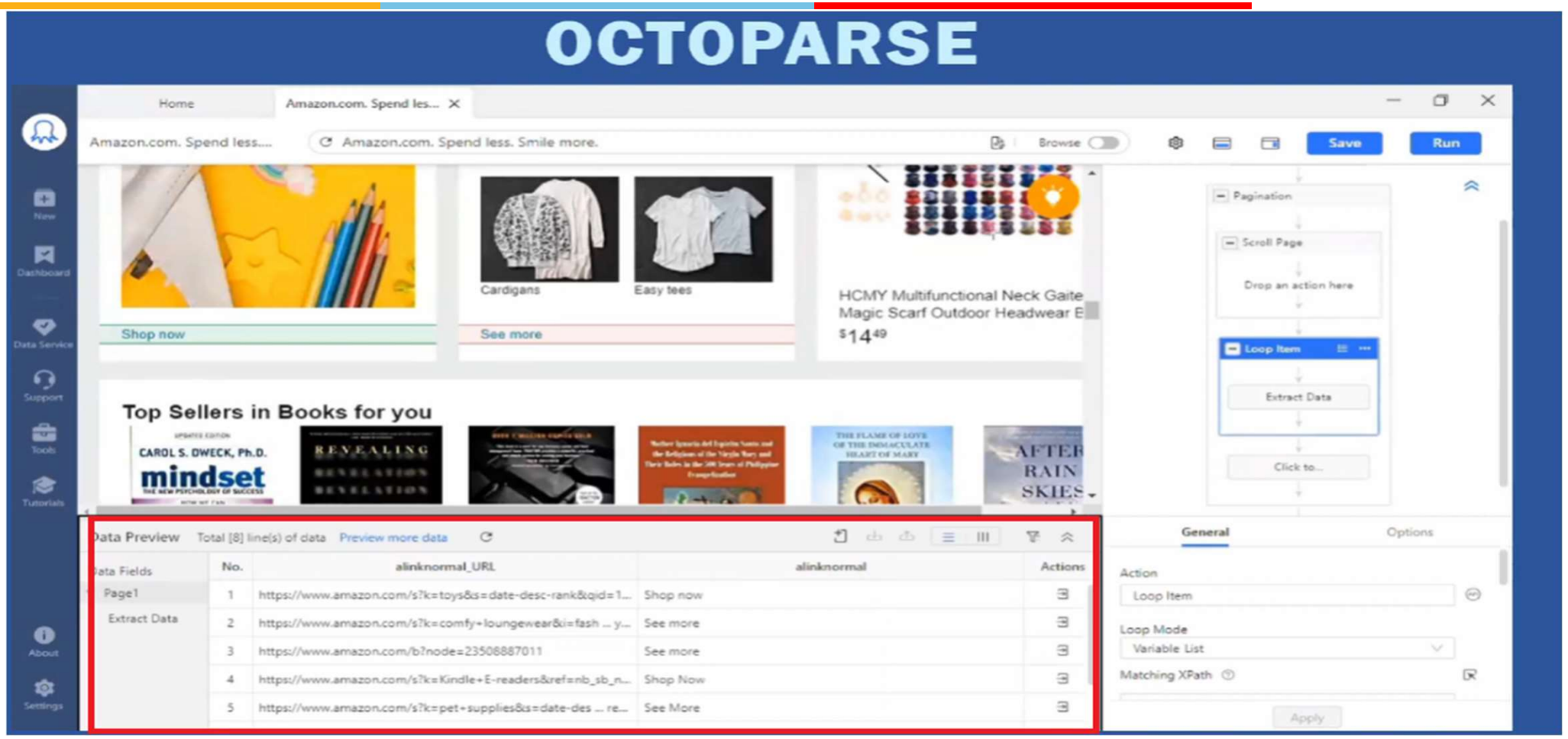
Avoid joins


Category
SubCategory
Category
ProductCode
Product Code
SubCategory
Items
Item
Product Code
DeNormalized
Item
Product Code
SubCategory
Category
Avoid SQL Joins ...They are Expensive

ETL Process


EXTRACTING
TRANSFORMING
LOADING
Processing or Transforming if needed
Change data as per required stucture,
Programatically
Remove Duplicate data and
Missing Information making
it usable and reliable
Supply converted data to
target location for storage
and analysis

Data Extraction tool-Example



Data Ingestion

Data Ingestion is the process of importing and loading data into a system. It's one of the most critical steps in any data analytics workflow. A company must ingest data from various sources.

Data ingestion and ETL are two very different processes. Data ingestion is importing data into a database or other storage engine, while ETL is extracting, transforming, and loading.

Data extraction is the first step in the data ingestion process, which involves pulling data from sources and preparing it for use.

Extraction liberates data so you can fully use its potential. Data ingestion transforms information into insight

Difference

Data Ingestion

Data ingestion is a process that involves copying data from an external source (like a database) into another storage location (like a database). In this case, it's typically done without any changes to the data.

For example, if you have an Amazon S3 bucket containing some files that need to be imported into your database, then data ingestion would be required to move those files into your database location.

ETL

ETL stands for extract transform load; it's a process that involves taking data from one system and transforming it so that it can be loaded into another system for use there.

In this case, rather than just copying data from one location to another without making any changes.

Ingestion and Integration

1) Data Ingestion - The act or process of introducing data into a database or other storage repository. Often this involves using an ETL (extract, transform, load) tool to move information from a source system (like Salesforce) into another repository like SQL Server or Oracle.

2) Data Integration - The process of combining multiple datasets into one dataset or data model that can be used by applications, particularly those from different vendors like Salesforce and Microsoft Dynamics CRM.

Ingestion Types

Real-time ingestion involves streaming data into a data warehouse in real-time, often using cloud-based systems that can ingest the data quickly, store it in the cloud, and then release it to users almost immediately.

Batch ingestion involves collecting large amounts of raw data from various sources into one place and then processing it later. This type of ingestion is used when you need to order a large amount of information before processing it all at once.

Ingestion process

Data ingestion process begins by prioritizing data sources, validating individual files and routing data items to the correct destination

1)Collection of Data from the source

2)Filtering

3)Route to one or more data stores

Ingestion challenges

1. **Coding and maintenance** are two enormous challenges that can take time to overcome. Sometimes it's easier to throw out old data than figure out how to organize it so that you can use it for future projects.
2. **Latency** is another challenge companies face when trying to ingest new data. If you're waiting too long between ingesting your data and using it in another application or process, then there may be significant delays in getting things done!
3. **Data quality** is also a challenge—how often have you had to clean up or reprocess old data because there wasn't enough information or detail? Sometimes we'll even need to go back through old files multiple times before they're ready for our purposes!
4. Finally, there's the problem of **capturing all this information** in the first place—how do we even begin collecting all this data without losing any of its required information?

Parallelism

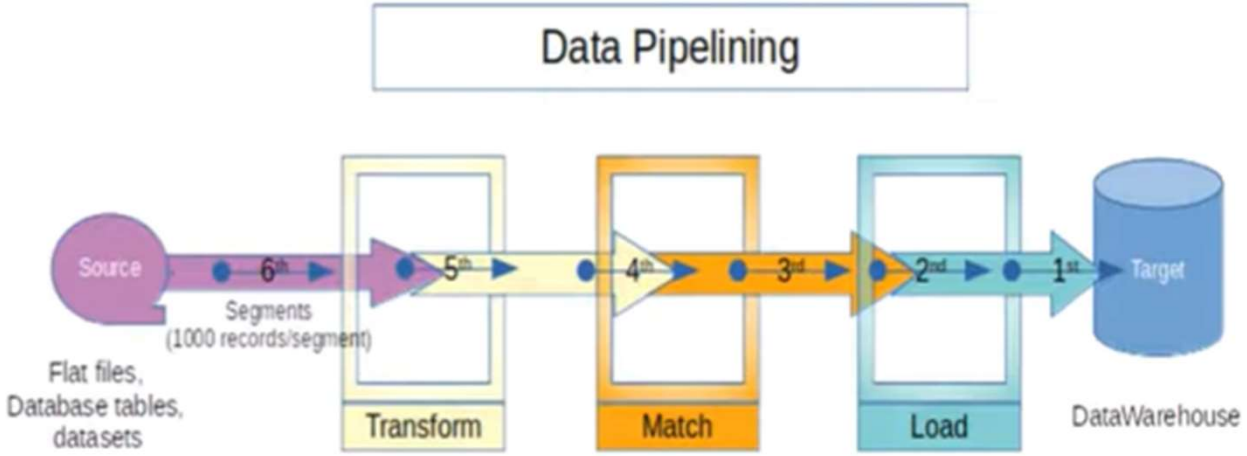
Methods of Parallelism

- Data Pipelining
- Data Partitioning
- Combining both Pipelining and Partitioning
- Dynamic Data Repartitioning

Pipelining

Data Pipelining

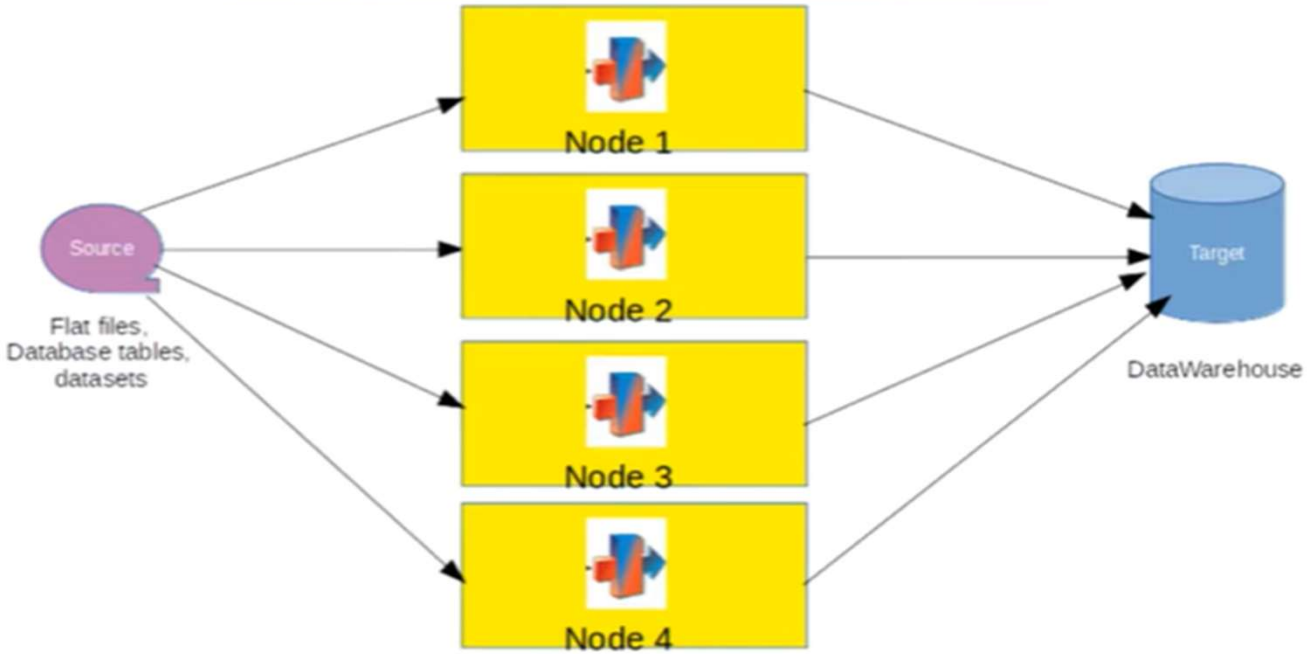
Entire data is partitioned and each partition Data will be Moving from one stage to next stage.



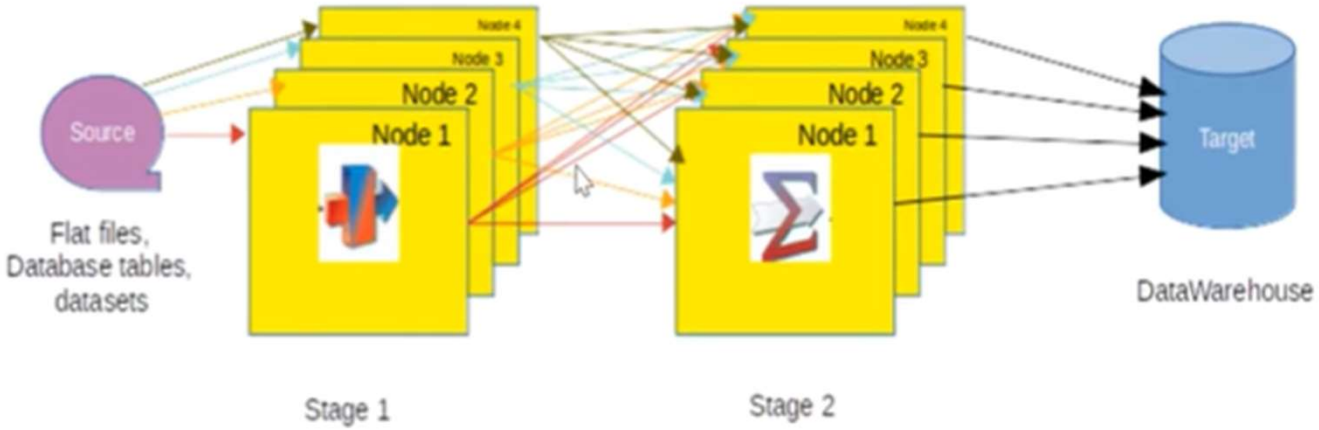
Partitioning

Data Partitioning Parallelism

All Stages will be running parallelly. Each note will have all stages running parallelly



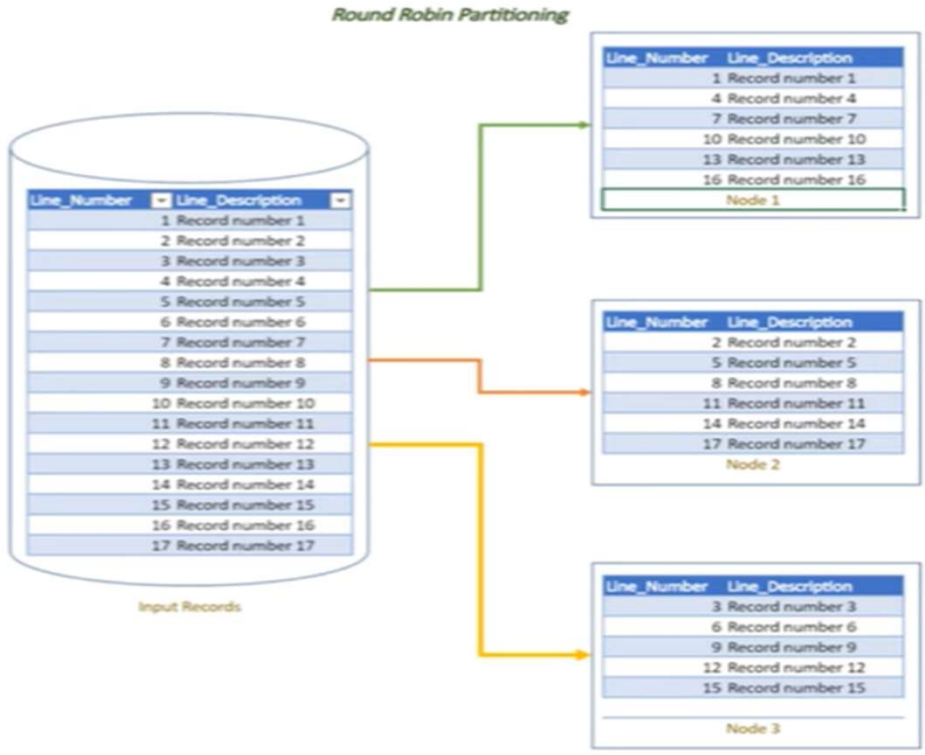
Dynamic Data Repartitioning



Data Clubbed based on customer last name

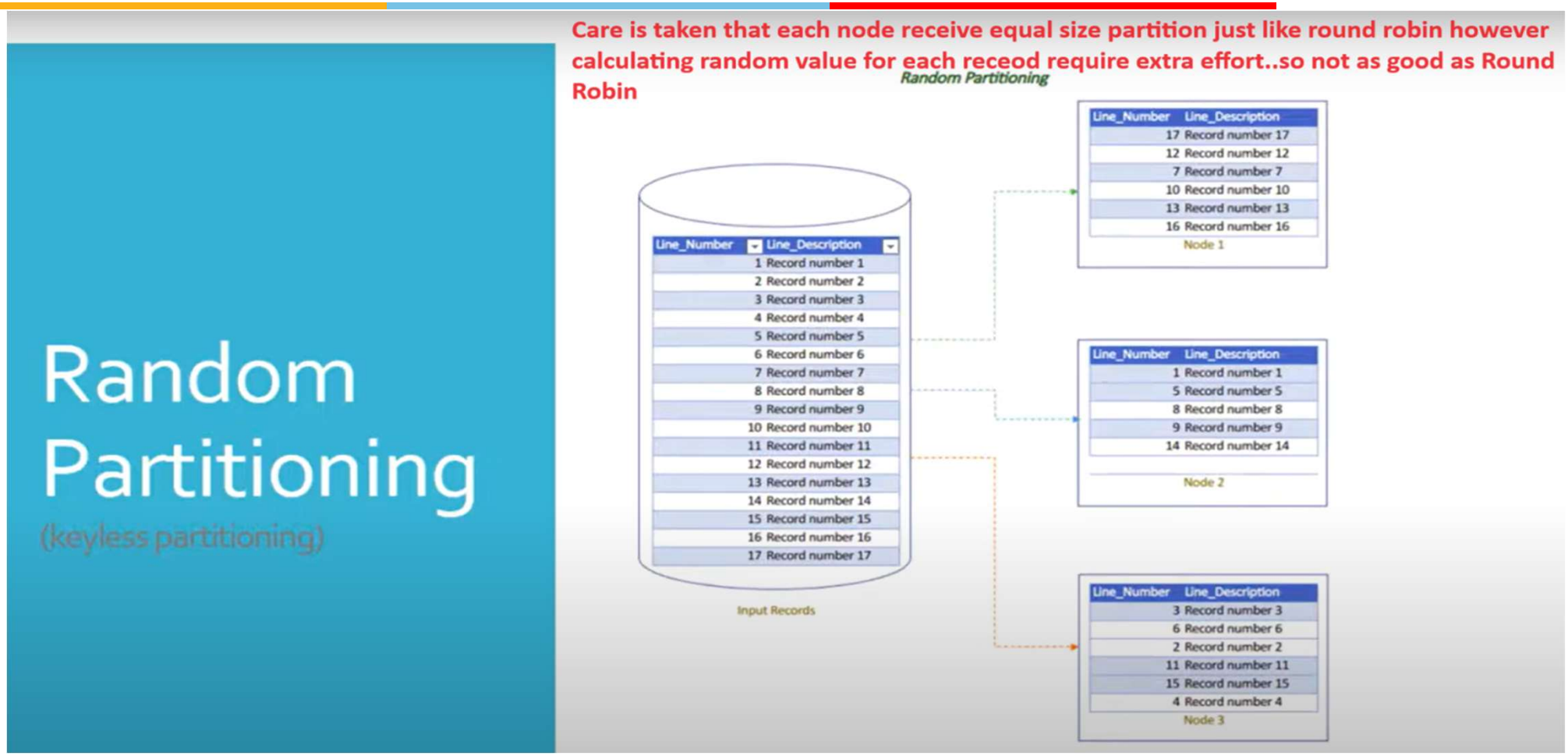
Data Partitioned Based on City

Round Robin Partition



Approximately equal sized partitions are made

Random Partitioning



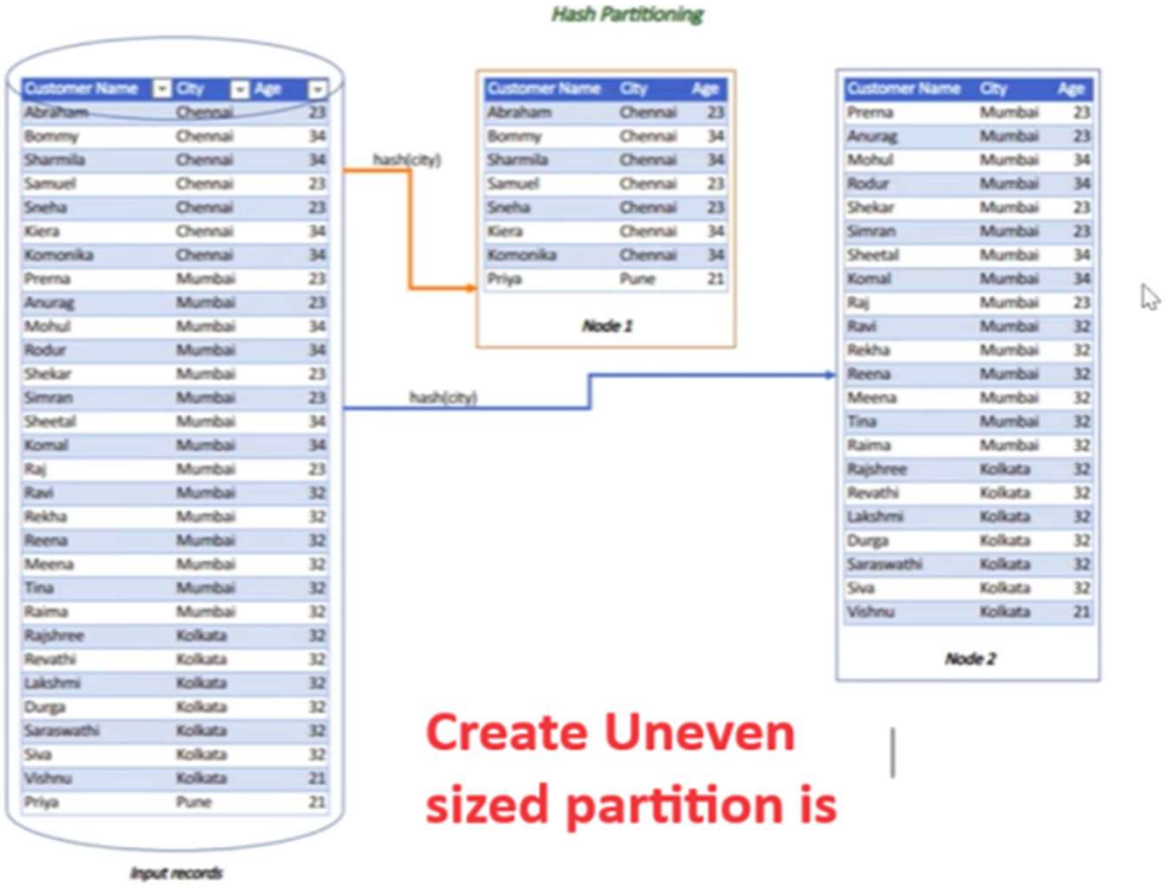
Same Partitioning

Mostly Used to pass data between two stages of processing

Partition and Divide-and-conquer techniques in Distributed system

Hash Partitioning

(key based)

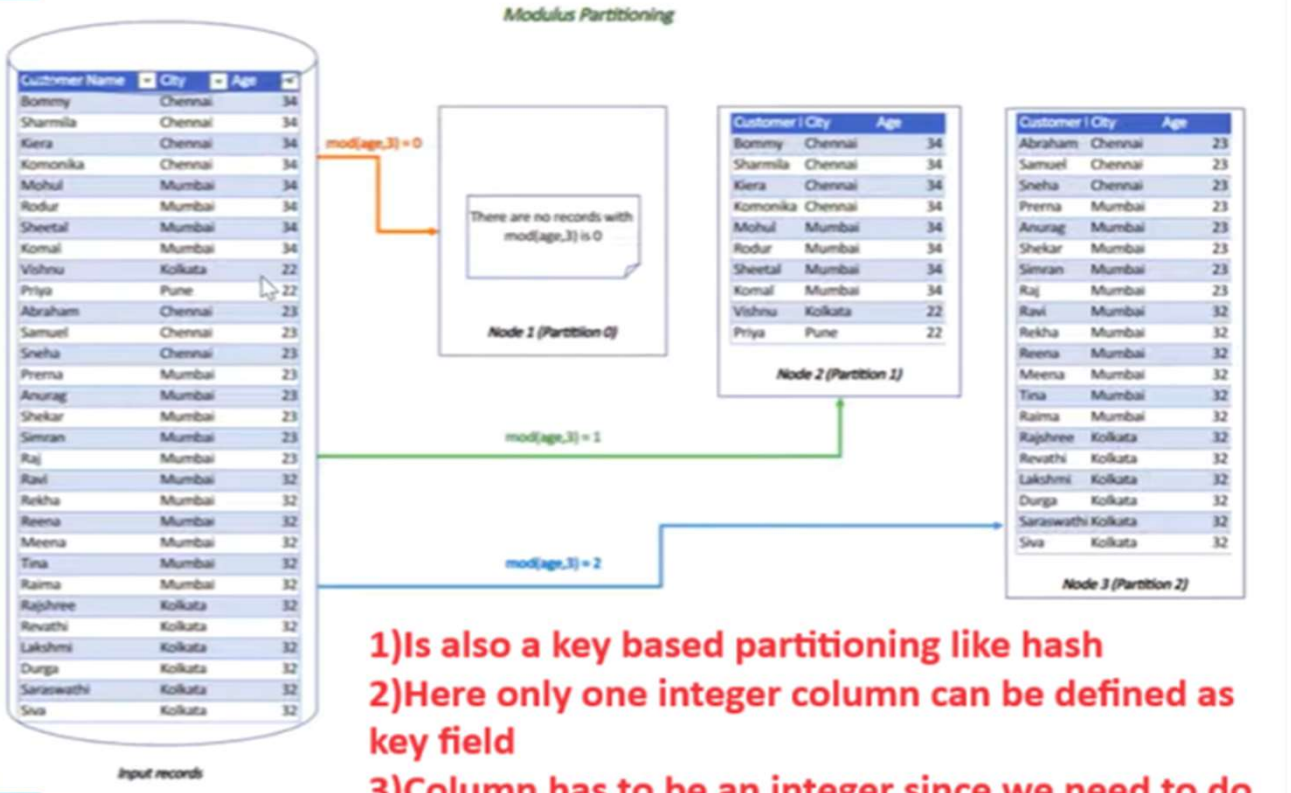


Modulus Partition

innovate achieve lead

Modulus Partitioning

(key based)

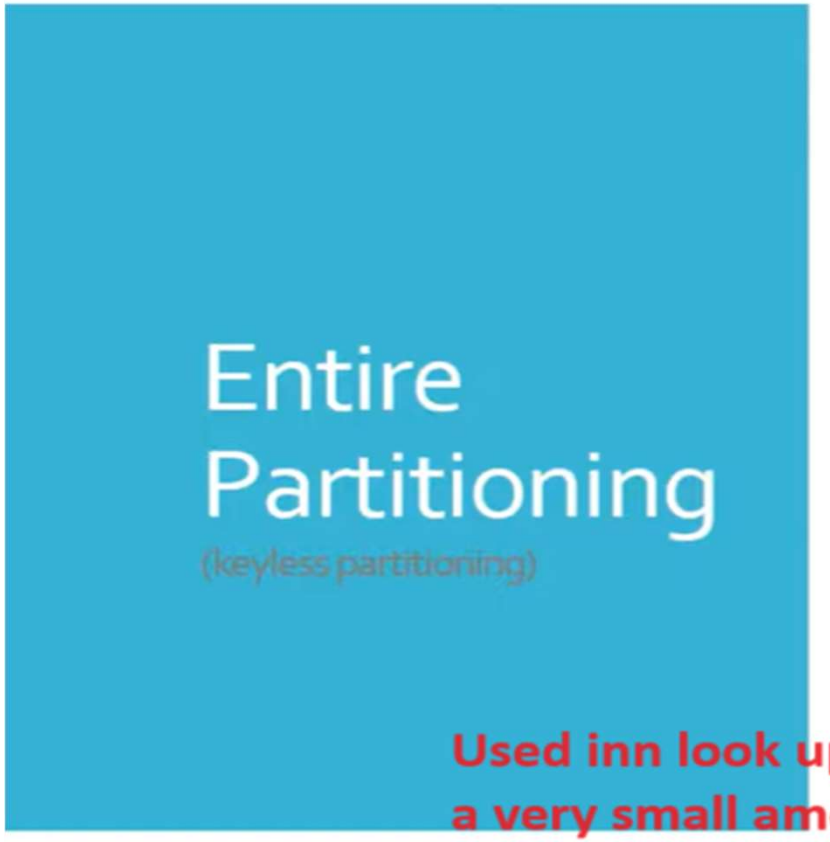


1)Is also a key based partitioning like hash

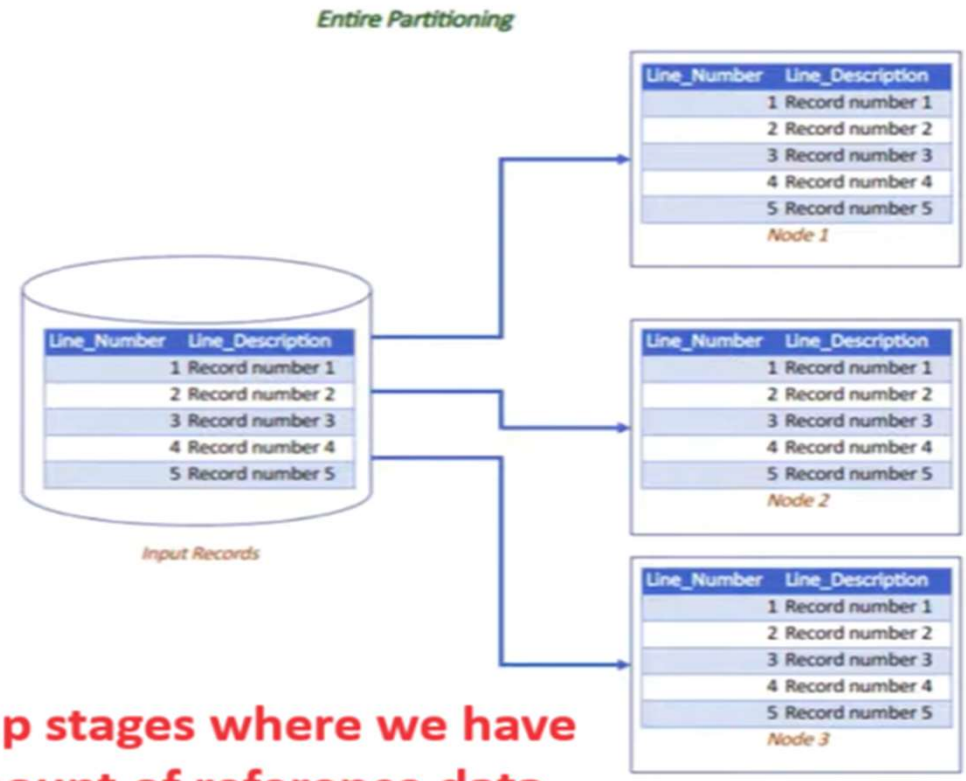
2)Here only one integer column can be defined as key field

3)Column has to be an integer since we need to do modulus operations to define partition

Entire Partitioning



Data is copied entirely in all nodes



Used inn look up stages where we have a very small amount of reference data

In-node and over-the-network latencies

In-Node Latency: In-node latency refers to the time it takes for a process or thread to access data or perform computation within the local memory of a single node. In-node latency is typically much lower compared to network latencies since data can be accessed and processed directly from local memory, avoiding the communication overhead associated with network transmission.

Over-the-Network Latency: Over-the-network latency refers to the time it takes for data to be transmitted between nodes over the network. Network latencies are typically higher than in-node latencies due to factors such as network congestion, routing delays, and transmission times.

The difference between in-node and over-the-network latencies is crucial in distributed systems, as it determines the cost of data communication and influences the design of communication patterns between nodes. Minimizing over-the-network latencies is essential to achieve better performance and responsiveness in distributed systems.

Data and Task Locality

Data Locality: Data locality refers to the degree to which data accessed by a process or task is physically located close to the processing unit (CPU, GPU) that needs it. High data locality means that the data accessed is already present in the local memory of the processing unit, reducing the need for expensive network communication or disk I/O. Data locality is vital in reducing access latencies and optimizing data processing in distributed systems.

Task Locality: Task locality refers to the placement of related computation or tasks close to each other, often within the same node. Task locality can improve performance by reducing communication costs between tasks, as communication within a node is faster than communication between nodes.

In distributed systems, optimizing data and task locality helps to reduce communication overhead and latency, leading to improved system performance and efficiency.

Communication cost

Communication cost refers to the resources, time, and bandwidth required to exchange data or messages between nodes in a distributed system. Communication costs include both the time taken to transmit data over the network (network latency) and any additional processing overhead required for serialization, deserialization, and handling communication protocols.

Minimizing communication cost is essential for achieving efficient data exchange and coordination among distributed nodes. Strategies such as data partitioning, data replication, and message aggregation can help reduce communication overhead and optimize data access and processing in distributed systems.

Conclusion

Efficient management of these factors can lead to better performance, reduced latency, and improved resource utilization in distributed computing environments.

Thank you

That is all