# CHAPTER 1

DATA WAREHOUSE FEATURES | INTRODUCTION

<https://www.javatpoint.com/data-warehouse>

According to the definition of Bill Inmon – Data Warehouse is a subject oriented, integrated, non-volatile and time variant collection of data in support of management's decision.”

Subject-Oriented Data: In Data Warehouse, data is stored by subjects, not by application.

Integrated Data: Data in data warehouse comes from several operational systems.

Time-Valiant Data: It means Data Warehouse has to Contain Historical data, not just Current Values. Every data structure in the data Warehouse Contains the time element. This feature allows the following: 1. Allows the analysis of Past. 2. Relates Information to the Present. 3. Enables forecast to future.

Non-Volatile Data: Data is not updated/ deleted from Data Warehouse in Real Time.

Data Granularity: In Data Warehouse, it is efficient to Keep date Summarized at different levels.

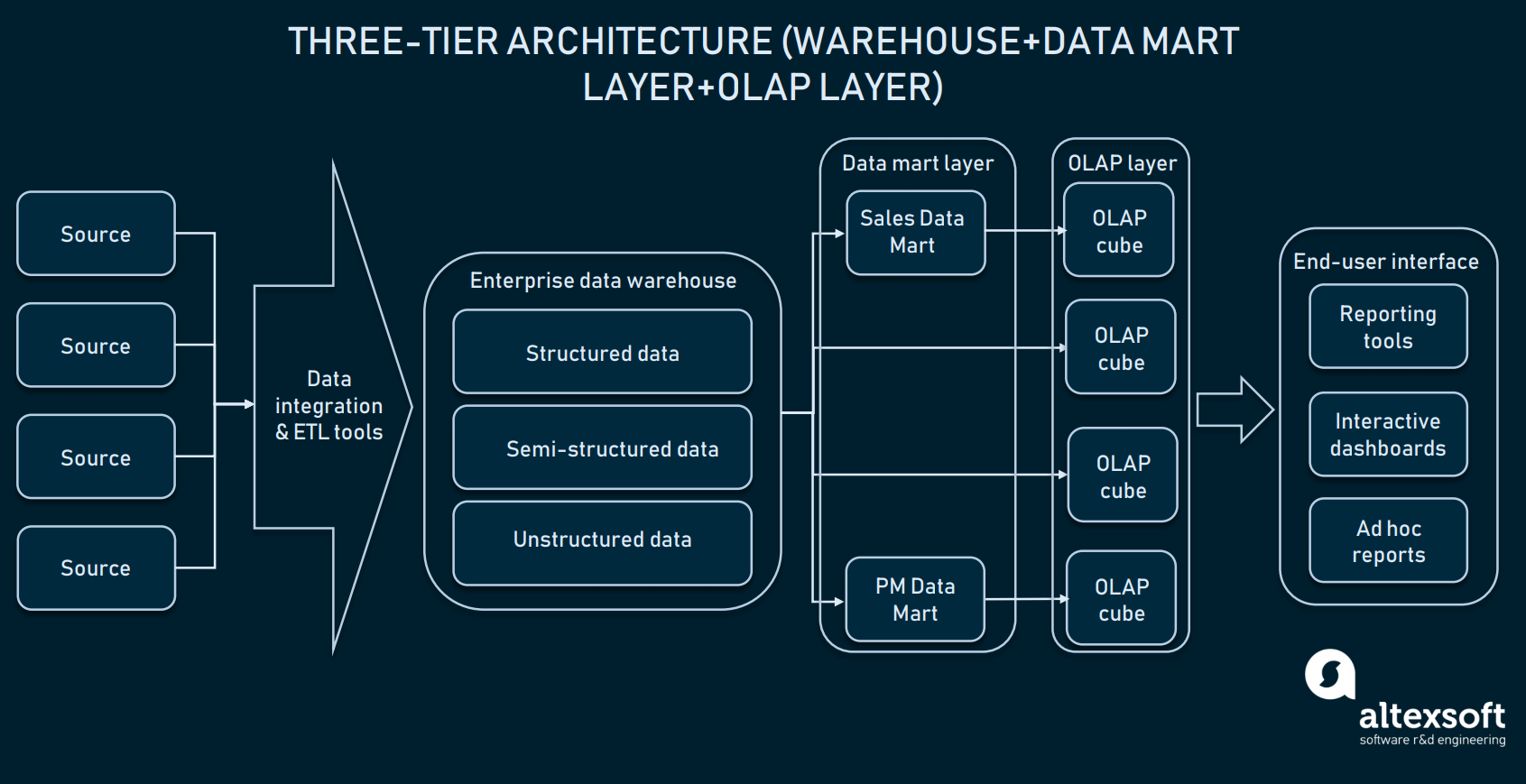
DATA WAREHOUSE COMPONENTS | 3 LAYER ARCHITECTURE

Data Warehouse Components – <https://www.javatpoint.com/data-warehouse-components>

Data Warehouse Architecture – <https://www.javatpoint.com/data-warehouse-architecture>

Three-Tier Data Warehouse Architecture – <https://www.javatpoint.com/three-tier-data-warehouse-architecture>

The goal of data warehouse architecture is to design a system that can meet the data needs of the organization in a secure, scalable, and performant way.



NEED FOR DATA WAREHOUSE

<https://youtu.be/WgIbvkyY4mI?si=kVtPV81bygJF_TpE>

ETL PROCESS

<https://www.javatpoint.com/etl-process-in-data-warehouse>

Extraction: Involves connecting to source system, and both selecting and collecting necessary data needed for analytical processing.

Transformation: Series of steps are performed on extracted data to convert into standard format.

Loading: Imports transformed data into a large database or data warehouse.

ETL VS ELT - <https://www.javatpoint.com/data-warehouse-etl-vs-elt>

DISTRIBUTED DATA WAREHOUSE

In a distributed data warehouse, data is not stored in a single server but is instead distributed across multiple servers or nodes.

Here are some examples of distributed data warehouse technologies:

• Amazon Redshift

• Google BigQuery

For example, Netflix uses Amazon Redshift to analyze its customer data. Netflix has over 200 million subscribers worldwide, and it generates a huge amount of data every day. Amazon Redshift allows Netflix to quickly and easily analyze this data to gain insights into its customers' viewing habits and preferences.

Another example of a distributed data warehouse is the Airbnb data warehouse. Airbnb uses Amazon Redshift to store and analyze its data on listings, bookings, and guests. This data allows Airbnb to improve its products and services, and to target its marketing campaigns more effectively.

TYPES OF DATA WAREHOUSES - <https://www.javatpoint.com/types-of-data-warehouses>

DATA MART INTROUDCTION

<https://www.javatpoint.com/data-warehouse-what-is-data-mart>

A data mart is a subset of a data warehouse that is focused on a specific business area or function. It is a smaller, more focused version of a data warehouse that is designed to be easier to use and manage for specific users.

STEPS IN IMPLEMENTING DATA MART

The steps in implementing a data mart are as follows

Design: This step involves gathering and analyzing business requirements, identifying data sources, and designing the logical and physical structure of the data mart.

Construction: This step involves building the data mart infrastructure, including the database and data loading process.

Population: This step involves loading data into the data mart from the source systems.

Access: This step involves developing tools and reports to allow users to access and analyze the data in the data mart.

Management: This step involves ongoing monitoring and maintenance of the data mart, including refreshing the data, optimizing performance, and managing user access.

DIMENSION MODELLING

Dimensional Modeling - <https://www.javatpoint.com/data-warehouse-what-is-dimensional-modeling>

Multi-Dimensional Data Model - <https://www.javatpoint.com/data-warehouse-what-is-multi-dimensional-data-model>

Data Cube - <https://www.javatpoint.com/data-warehouse-what-is-data-cube>

Dimensional modeling is just a technique to create a data warehouse or a database for reporting and analytical purposes by dividing the data into measure and dimension.

Suppose you are working for a retail company, and you want to analyze sales data. You need to create a dimensional model for this data. Dimensional model is made of Fact Table and Dimension Table

**Fact Table: Sales**

The fact table is where the numerical data (measures or metrics) are stored. In this case, the central fact table would contain information about sales.

DateKey: A foreign key that links to the Date dimension table.

ProductKey: A foreign key that links to the Product dimension table.

StoreKey: A foreign key that links to the Store dimension table.

SalesAmount: The measure, which represents the total sales revenue for that transaction.

Here's what the fact table might look like:

|  |  |  |  |
| --- | --- | --- | --- |
| DateKey | ProductKey | StoreKey | SalesAmount |
| 202301 | 101 | 501 | 500.00 |
| 202302 | 102 | 502 | 750.00 |
| 202302 | 103 | 501 | 300.00 |

**Dimension Tables: Date, Product, Store**

Dimension tables provide context and descriptive information about the data in the fact table. They typically contain attributes and hierarchies that you can use for slicing and dicing the data. Here's a brief overview of the three dimension tables:

Date Dimension Table:

DateKey

Date

Day

Month

Quarter

Year

Product Dimension Table:

ProductKey

Product Name

Category

Brand

Store Dimension Table:

StoreKey

Store Name

Location

Manager

With this dimensional model in place, you can easily answer various business questions, such as:

"What were the total sales for each product category in Q1 2023?"

"How do sales in Store X compare to Store Y over the past three months?"

"What are the top-selling products in Store Z last year?"

In summary, dimensional modeling is a data modeling approach that organizes data into fact and dimension tables, making it easy to perform complex analysis

In general, dimensions are used to describe and filter the data, while measures are used to quantify and analyze the data.

STAR, SNOWFLAKE, FACT CONSTELLATION SCHEMA

A data warehouse schema is a logical description of the data in a data warehouse.

It defines the tables, columns, and relationships between the tables.

The schema is used to ensure that the data in the data warehouse is consistent and well-organized.

Star schema:

<https://www.javatpoint.com/data-warehouse-what-is-star-schema>

A star schema has a single fact table at the center, surrounded by dimension tables.

The fact table contains quantitative data, such as sales figures or customer counts.

The dimension tables contain descriptive data about the fact table, such as the customer's name or the product that was sold.

Snowflake schema:

<https://www.javatpoint.com/data-warehouse-what-is-snowflake-schema>

A snowflake schema is similar to a star schema, but the dimension tables are normalized.

This means that the dimension tables are broken down into smaller tables, which can improve performance and reduce data redundancy.

Fact constellation schema:

<https://www.javatpoint.com/data-warehouse-what-is-fact-constellation-schema>

A fact constellation schema is a more complex schema that consists of multiple fact tables and dimension tables.

The fact tables are related to each other through the dimension tables.

This schema is typically used to support complex analytical queries.

DATA WAREHOUSE SCHEMA KEYS

A primary key is a column or a set of columns in a table whose values uniquely identify a row in the table.

A foreign key is a column or a set of columns in a table whose values match the values of the primary key in another table. Foreign key relationships allow us to link data between tables.

A surrogate key is a column or a set of columns in a table whose values are generated by the database and are not derived from any other data in the table. Surrogate keys are often used as primary keys in data warehouses and data marts.

Here's an example to illustrate these concepts:

Suppose you have two tables: "Employees" and "Departments."

Employees Table:

Primary Key: EmployeeID (a surrogate key, an auto-incrementing integer)

Other columns: FirstName, LastName, DepartmentID (a foreign key)

Departments Table:

Primary Key: DepartmentID (a surrogate key, an auto-incrementing integer)

Other columns: DepartmentName

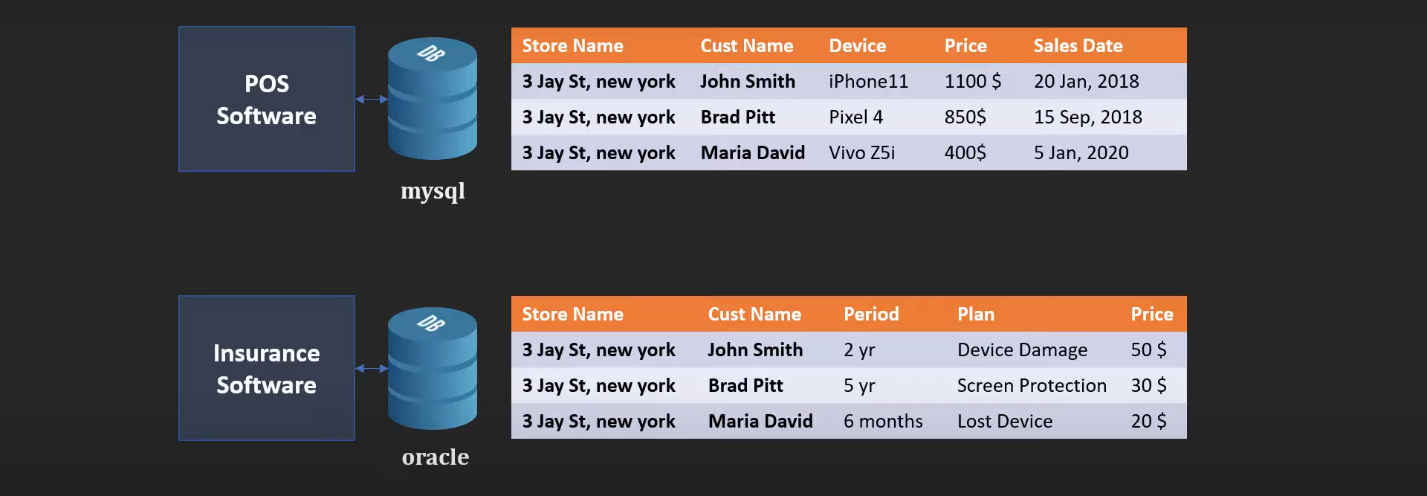
In this example, "EmployeeID" in the "Employees" table is a surrogate key that uniquely identifies each employee. It has no inherent meaning beyond being a unique identifier for the records in that table. The "DepartmentID" in the "Employees" table is a foreign key, linking each employee to their respective department via the "DepartmentID" in the "Departments" table. "DepartmentID" in the "Departments" table is also a surrogate key, serving as the primary key for that table.

OLAP INTRODUCTION

Imagine a store of Vijay sales or Croma. They have stores across the country. When you walk into the store, the store will have different software systems such as POS and insurance software.

The POS system tracks the sales transactions. When people buy phones, they generally buy insurance as well and to track the insurance the store will use insurance software which will have insurance related records in it.

These two systems might be using different databases. For example, your POS system might be using MySQL whereas your insurance system might be using Oracle.



Other than that, you might have unstructured data for example, you are running customer satisfaction survey where you are scanning those survey papers and storing those as PDFs on your Amazon s3 cloud.

Now, in your business you have the structured as well as unstructured data and as a business owner now you want to get insights into your business. You might want to ask different questions such as which store is performing best in terms of device and insurance sales total? In terms of customer satisfaction which store and employee ranks the best? Holiday season is coming, which region is going to have maximum traffic of customers?

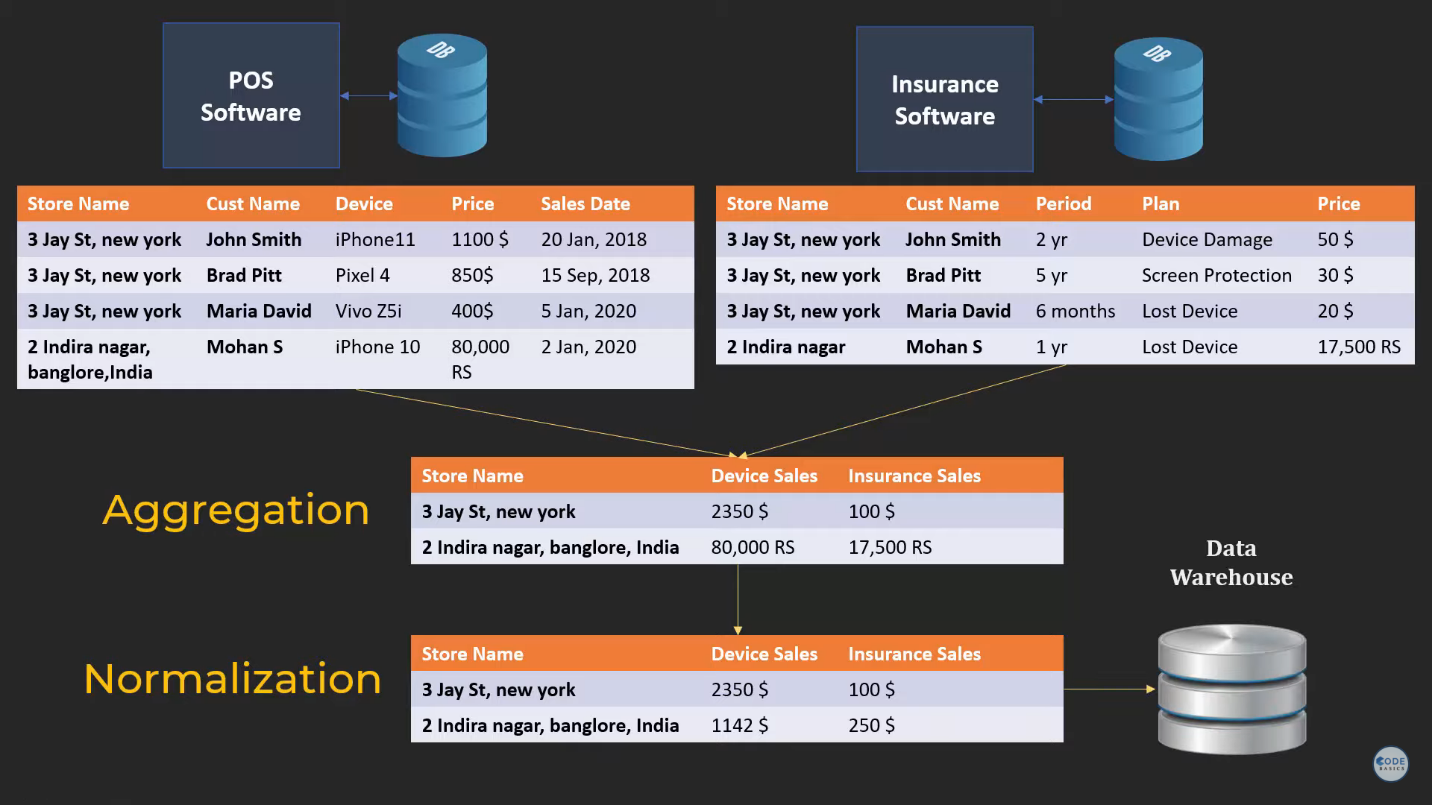
Nowadays getting insights from the data has become very important because this kind of insights can help you make better business decision and future business strategy.

Let’s take the first question which is which store is performing best in terms of device and insurance sales total is performing best. Now one obvious thing you can do is you have these two databases one is MySQL one is Oracle you can get the aggregate sales numbers from the first one and then aggregate insurance number from the second one and you can just add them up but this is not a good idea because these two are mission critical systems.

These systems are serving your real customers. The query that your forming, what if that query is complex and it slows down the database? That's not going to be good.

So typically, what organizations do is they copy all of this data into some other database and they perform the query on that database. Because even if you slowdown that database, it is not serving your customers. So, it's going to be okay.

Now you don't want to blindly copy these records into the second database that we just talked about because to make your analytics easy you might want to apply some transformation.

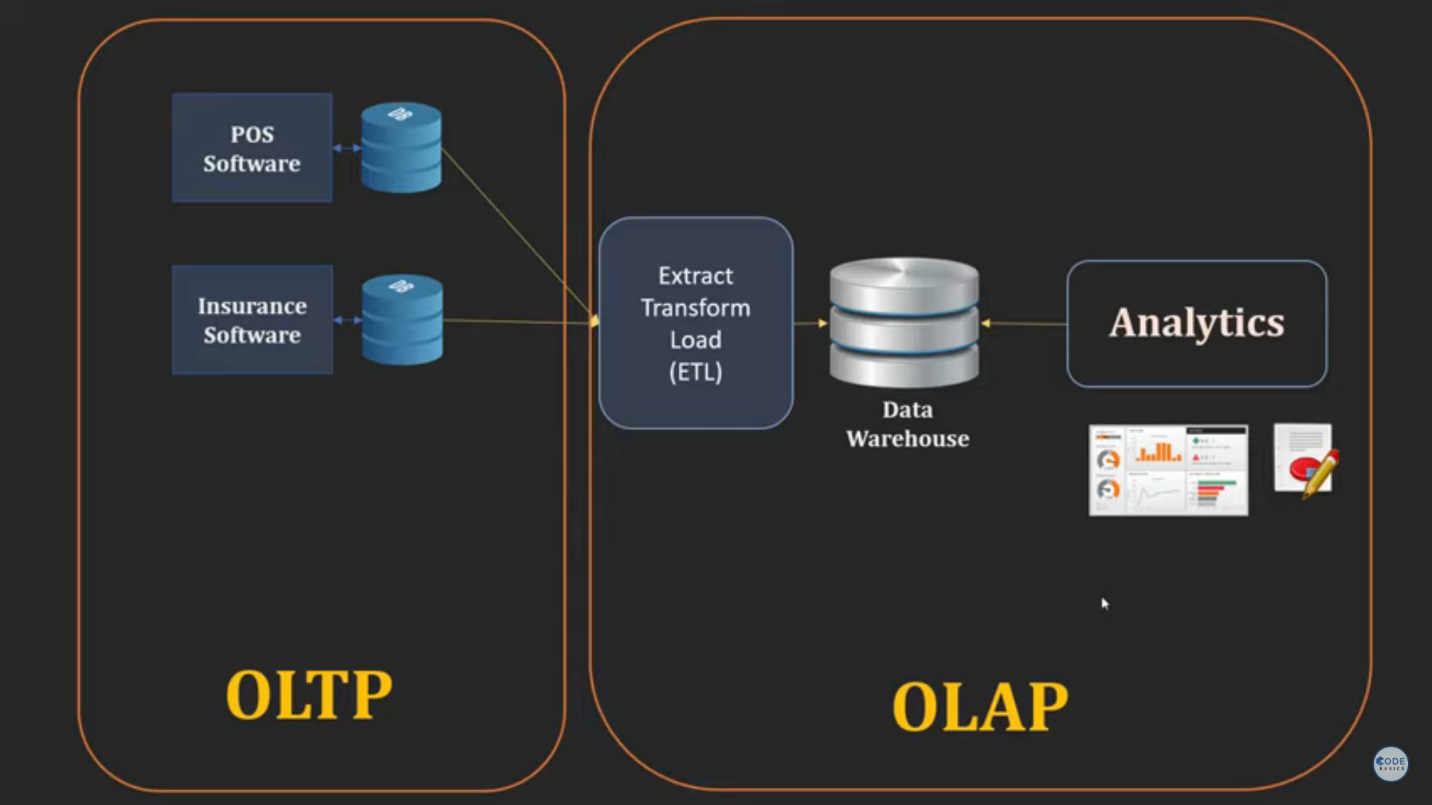


Once I have the records in a format that I can perform my queries on I can then store them into another database called a data warehouse.

Basically, all kinds of data that is being generated in your organization they all can be dumped into this data warehouse after applying some transformation so that later on you can perform analytics on top of it.

If you realize we performed three steps. We first extracted data from various sources so that step is called extraction then we did an aggregation and normalization so that step is called transformation. After that we loaded these records into data warehouse that step is called load hence ETL extraction!

After performing ETL you store it into data warehouse where data analyst and data scientist will perform various queries and analysis to answer the questions we asked before.



The left-hand side box is called OLTP which is online transaction processing system and these two databases are called OLTP databases.

The right-hand side box is called OLAP which is online analytical processing system and the data warehouse here is called OLAP database.

The OLTP is your transactional databases which are serving your main business, which are serving your customers. These are mission critical system and you want to make sure they don't slow down. They are up all the time.

NOW OLTP on the other hand is something that you need in house. It is not solving to your customers but you need it getting insight from your business.

So, in summary,

OLAP (Online Analytical Processing) is a system that supports complex data analysis of large amounts of historical data. OLAP systems are typically used by business analysts to make decisions about the future of a business.

OLTP (Online Transaction Processing) is a system that supports high-volume, real-time data processing of online transactions. OLTP systems are typically used by businesses to process customer orders, manage inventory, and update customer accounts.

Examples of OLAP systems include:

Google BigQuery

Amazon Redshift

Microsoft Azure Synapse Analytics

Snowflake

Examples of OLTP systems include:

MySQL

PostgreSQL

Oracle Database

Microsoft SQL Server

OLAP OPERATIONS

<https://www.javatpoint.com/olap-operations>

Drill-down: Drill-down allows users to move from a summary level of data to a more detailed level. For example, a user could drill down from total sales by product category to total sales by product subcategory.

For example, we might start by looking at total sales for the electronics category. We could then drill down to total sales for the television subcategory. Finally, we could drill down to total sales for the Samsung 4K TV product

Roll-up: Roll-up is the opposite of drill-down. It allows users to move from a detailed level of data to a summary level. For example, a user could roll up from total sales by product subcategory to total sales by product category.

For example, we could start by looking at total sales for individual products. We could then roll up to total sales by product subcategory. Finally, we could roll up to total sales by product category.

Slice and dice: Slice and dice allow users to view different subsets of data. For example, a user could slice the data by region to see how sales are performing in different parts of the world. Or, the user could dice the data by time to see how sales have changed over time.

For example, we could slice the data by region to see how sales are performing in different parts of the world. Or, we could dice the data by time to see how sales have changed over time.

Pivot: Pivot allows users to change the orientation of the data in the cube. For example, a user could pivot the data to see sales by product category for each region.

For example, we might pivot the data so that we can see sales by product category for the North America region. This would allow us to see which product categories are selling well in North America.

ROLAP, MOLAP & HOLAP

<https://www.javatpoint.com/rolap-vs-molap-vs-holap>

ROLAP stands for Relational Online Analytical Processing. It is a type of OLAP that uses a relational database to store and manage data. The data is stored in tables with relationships between them.

MOLAP stands for Multidimensional Online Analytical Processing. It is a type of OLAP that uses a multidimensional database to store and manage data. Fact tables and Dimension tables.

HOLAP stands for Hybrid Online Analytical Processing. It is a type of OLAP that combines the best features of ROLAP and MOLAP. HOLAP systems typically store aggregated data in a multidimensional database, but they also allow users to access and analyze detailed data in the relational database.

# CHAPTER 2

DATA MINING INTRODUCTION

<https://www.javatpoint.com/data-mining>

It is the process of discovering or mining knowledge from a large amount of data.

Another Term for Data Mining is KDD (Knowledge discovery from data)

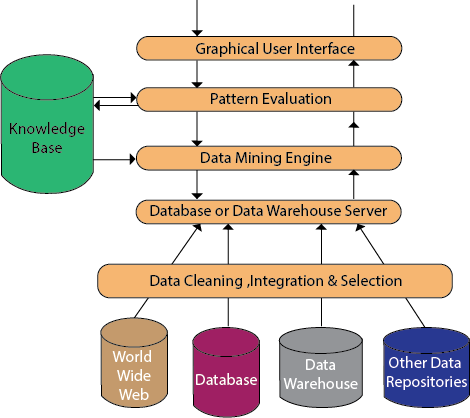
Attempts to extract hidden patterns and trends from large databases.

In gold mining, miners sift through large amounts of dirt and rock to find small nuggets of gold. In data mining, analysts sift through large amounts of data to find small patterns and insights.

Another analogy for data mining is treasure hunting. Treasure hunters use maps and clues to find buried treasure. Data miners use algorithms and statistical methods to find hidden patterns in data.

DATA MINING ARCHITECTURE

<https://www.javatpoint.com/data-mining-architecture>



STEPS IN DATA MINING PROCESS

Data preparation: The analyst would clean the data to remove errors and inconsistencies. They would also integrate data from multiple sources, such as the credit card company's database and the customer's fraud history. Finally, they would convert the data into a format that is compatible with the data mining algorithm.

Data modeling: The analyst would select a data mining algorithm that is appropriate for fraud detection. For example, they might select a decision tree or support vector machine algorithm. They would also set the parameters of the algorithm, such as the minimum number of samples required to create a node in the decision tree.

Data mining: The analyst would run the data mining algorithm on the prepared data. The algorithm would generate a set of patterns or insights that can be used to identify fraudulent transactions. For example, the algorithm might identify patterns in the transaction amounts, locations, or times that are associated with fraud.

Pattern evaluation: The analyst would evaluate the patterns or insights generated by the data mining algorithm to assess their usefulness and validity. The analyst might also test the patterns or insights on a held-out test dataset to see how well they perform in a real-world setting.

Knowledge representation: The analyst would present the knowledge extracted from the data in a way that is understandable and useful to the credit card company. For example, the analyst might create a report that identifies the most common types of fraud, or they might develop a fraud detection model that can be used to flag suspicious transactions.

DATA MINING TASKS

<https://www.javatpoint.com/tasks-and-functionalities-of-data-mining>

Predictive:

Predict Values by making of known results from a different set of sample data.

Classification: Classifying data into different categories. For example, classifying emails as spam or not spam.

Regression: Predicting a continuous value. For example, predicting the price of a house.

Time series forecasting: Predicting future values of a time series. For example, predicting the future sales of a product.

A bank might use predictive data mining to predict which customers are most likely to default on a loan. A retailer might use predictive data mining to predict which products are most likely to sell well during the holiday season.

Descriptive:

Enables to determine you patterns and relationship in a Sample data.

Clustering: Grouping similar data points together. For example, clustering customers into different segments based on their purchase history.

Association rule mining: Finding patterns in data that co-occur frequently. For example, finding products that are often purchased together.

Summarization: Aggregating data to provide a high-level overview. For example, calculating the average sales for each product category.

A hospital might use descriptive data mining to identify patients who are at risk of developing a particular disease. A marketing agency might use descriptive data mining to identify customer segments that are most likely to be interested in a new product.

DATA PROCESSING IN DATA MINING

<https://www.javatpoint.com/data-processing-in-data-mining>

MAJOR DATA MINING TECHNIQUES:

K-Means Clustering

<https://www.javatpoint.com/k-means-clustering-algorithm-in-machine-learning>

<https://www.youtube.com/watch?v=CLKW6uWJtTc>

Hierarchical Clustering

<https://www.javatpoint.com/hierarchical-clustering-in-machine-learning>

Decision Tree

<https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm>

<https://www.youtube.com/watch?v=RVuy1ezN_qA>

Association Rule Learning

<https://www.javatpoint.com/association-rule-learning>

FP Growth Algorithm

<https://www.javatpoint.com/fp-growth-algorithm-in-data-mining>

<https://youtu.be/GzRi6pbdQ2E?si=ZQZQ7mHfYY6xMl5f>

Naïve Bayes

<https://www.javatpoint.com/machine-learning-naive-bayes-classifier>

<https://youtu.be/mzPHmNm_NrM?si=MMdMgrhBU_T4jMaP>

Linear Regression

<https://www.javatpoint.com/linear-regression-in-machine-learning>

<https://youtu.be/lzGKRSvs5HM?si=BiKhxY2RJPAHdnx6>

Logistic Regression

<https://www.javatpoint.com/logistic-regression-in-machine-learning>

<https://youtu.be/VImxF-9jk1E?si=e7Kw35bUEu19PD_p>

K-Nearest Neighbor(KNN)

<https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning>

<https://youtu.be/CJjSPCslxqQ?si=zrfW4zkuahXeImzP>

ANN

<https://www.javatpoint.com/artificial-neural-network>

<https://youtu.be/kLaYXKqJBFw?si=Hv0PnS9by6ifz-kd>

# CHAPTER 3

BASIC DEFINITION OF BIG DATA | INTRODUCTION

<https://www.javatpoint.com/what-is-big-data>

Big data is a term used to describe the collection and analysis of datasets that are so large and complex that they cannot be processed using traditional data processing applications.

Big data can come from a variety of sources, including social media, sensors, financial transactions, and so on.

In other words, big data is the data of huge volume, high velocity and of different variety which is used for enhanced business insights and decision making.

This is also known as the three Vs of big data

**Volume** refers to the sheer amount of data that is collected and analyzed. Big data datasets can be petabytes or even exabytes in size. This volume of data is too large for traditional data processing applications to handle.

**Velocity** refers to the speed at which data is collected and processed. Big data datasets are often generated in real time, such as social media posts or sensor data. This means that businesses need to be able to analyze data quickly in order to make timely decisions.

**Variety** refers to the different types of data that are collected and analyzed. Big data datasets can include structured data (such as customer records or financial transactions), semi-structured data (such as emails or log files), and unstructured data (such as images, videos, and social media posts). This variety of data can be challenging to analyze using traditional data processing tools.

Here is an example of how the three V's of big data can impact a business:

A social media company collects data on user activity, such as the posts they like and share, the people they follow, and the ads they click on.

This data is collected in real time (velocity) and can be very large in volume.

The data is also very varied, including structured data (such as user profiles and post timestamps), semi-structured data (such as post text and comments), and unstructured data (such as images and videos).

The social media company can use this data to analyze user trends, develop new features, and target advertising.

However, the company needs to have the right technology and infrastructure in place to manage and analyze this data.

The company also needs to have employees with the skills and expertise to analyze big data.

HISTORY OF BIG DATA

The origins of large data sets go back to the 1960s and '70s when the world of data was just getting started with the first data centers and the development of relational database

Around 2005, people began to realize just how much data users generated through Facebook, YouTube, and other online services. I think you have already heard about Hadoop which is nothing but an open-source framework created specifically for storing and analyzing the big data sets and solving the big data problems. The NoSQL also began to gain popularity during this time.

With the advancement of the Internet of Things (IoT), more and more objects and devices are connected to the internet.

While big data has come far, its usefulness is only just beginning. Cloud computing has expanded the possibility of big data even further.

DIFFERENCE BETWEEN BIG DATA AND HADOOP?

<https://www.javatpoint.com/what-is-hadoop>

Big data is nothing but a term, it's a concept. While on the other hand, hadoop is a technology which is implemented to handle the big data problems.

Hadoop was named after Doug Cutting's son's stuffed yellow elephant. Cutting chose the name because it was easy to pronounce and spell, and it was not used elsewhere.

In an interview, Cutting explained that he wanted to choose a name that was "meaningless" and "not used elsewhere" so that it would be easy to trademark.

HADOOP ARCHITECTURE

Hadoop contains three core components which are nothing but HDFS which is hadoop distributed file system which is a storage layer for Hadoop, the Mapreduce which is a processing layer and yarn which is a resource management layer.

HDFS

<https://www.javatpoint.com/hdfs>

You save movies, text files, photos, videos, and all other types of data on your laptop's local disk (such as D, E, or C). This involves utilizing the Windows file system, allowing you to perform actions like accessing, deleting, renaming, copying, and cutting the data. Familiar with these operations? Similarly, Hadoop, too, possesses its own file system known as HDFS (Hadoop Distributed File System). Now, the intriguing question arises: How does HDFS function?

It's a storage layer for Hadoop and it has a master slave architecture.

Master is nothing but a high-end machine (Name Node) whereas the slaves (Data Nodes) are a commodity hardware.

In HDFS data files are divided into multiple blocks preferably each block has a size of 128 MB or 256 MB by default.

For example, let's consider we have a 1 GB of file and if the block size is 128 MB, it will divide those into seven to eight blocks and it will store it in a distributed fashion on clusters of slave nodes.

On the master the metadata recites so it will have the address and all of the information of the blocks which are stored in the slave nodes.

So basically, HDFS has two demons (In Hadoop, "daemons" are services that run in the background and perform essential tasks) running for it which are name node and data node.

Name nodes run on the master machine which is a high-end hardware.

It is responsible for maintaining, monitoring, and managing the data nodes so it acts as a master in HDFS.

It records the metadata files like the location of blocks the file size and the permission allotted to that file.

In simple words, it closely works with a data node to store all the information of our data files.

The second component is a data node.

Data node runs on the slave machine which are nothing but commodity hardware which are easily replaceable and cheap to buy so it stores the actual big data files.

Our 1 GB file has 7 to 8 blocks so these seven to eight blocks will be stored in the data node in distributed manner.

Data node does all the groundwork from creating, replicating as well as deleting the blocks on the command of name node

To better comprehend what you've recently understood,

The master node in a Hadoop cluster is a single computer that is responsible for managing the file system and allocating resources to applications.

The data nodes are other computers in the cluster that store the actual data files.

The master node and data nodes can be any type of computer, but they are typically commodity servers.

Data nodes are different servers in which data is stored, and the name node is the main server that is used to access the data nodes.

Imagine a library. The librarian is like the name node. The librarian keeps track of the location of all of the books in the library. When a patron wants to borrow a book, they go to the librarian to get the location of the book. The patron then goes to the shelf where the book is located and borrows the book.

The bookshelves are like the data nodes. The bookshelves store the books in the library. The bookshelves are also organized in a way that makes it easy for the librarian to find the books.

MAPREDUCE

<https://www.javatpoint.com/mapreduce>

When you access the data stored in your drive in your computer it is process by CPU right? Just like this CPU for data processing Hadoop uses MapReduce.

When you want to access the data which is stored in your HDFS, MapReduce will come to use.

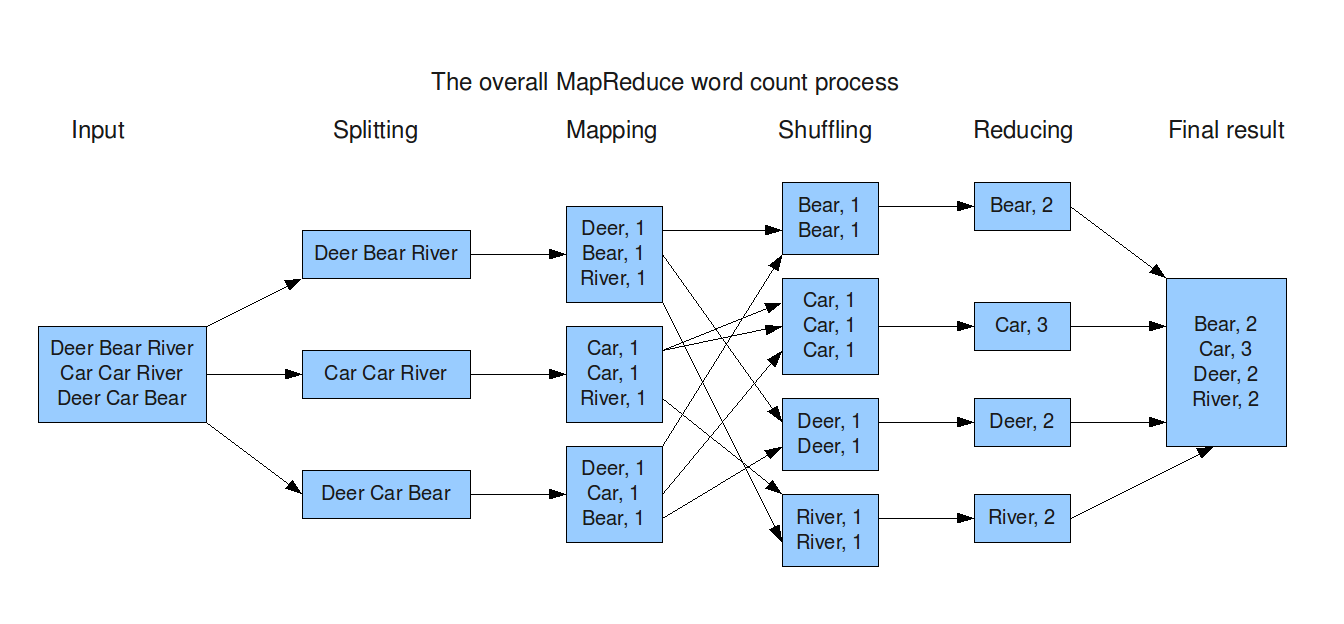
It breaks down a large job into smaller tasks that can be executed in parallel on multiple computers.

MapReduce consists of two main phases:

Map phase: The map phase splits the input data into key-value pairs and applies a mapping function to each pair. The mapping function transforms the input data into a format that is suitable for the reduce phase.

Reduce phase: The reduce phase takes the output of the map phase and combines it into a smaller set of key-value pairs. The reduce function combines the values for each key into a single value.

The map and reduce phases can be executed in parallel on multiple computers. This allows MapReduce to process large datasets quickly and efficiently.



Here is a simple example of how MapReduce can be used to count the number of words in a large text file:

Map phase:

The map phase splits the text file into lines and applies a mapping function to each line. The mapping function splits the line into words and generates a key-value pair for each word. The key is the word and the value is the count of the word.

Reduce phase:

The reduce phase takes the output of the map phase and combines it into a smaller set of key-value pairs. The reduce function sums the counts for each key to get the total count of the word.

Here is an example of how MapReduce can be used to access and process data in HDFS:

Suppose you have a large text file that contains a list of all the products that are sold on an e-commerce website. You want to use MapReduce to count the number of products that are sold in each category.

To do this, you would write a MapReduce job that reads the text file and splits it into lines. Each line would then be processed by the map function, which would extract the product category from the line. The map function would then emit a key-value pair, where the key is the product category and the value is 1.

The reduce function would then sum the values for each key to get the total number of products in each category. The results of the reduce function would then be written to HDFS.

Once the job is finished, you can use the results to analyze the product sales data. For example, you could use the results to identify the most popular product categories or to track product sales over time.

YARN

<https://www.javatpoint.com/yarn>

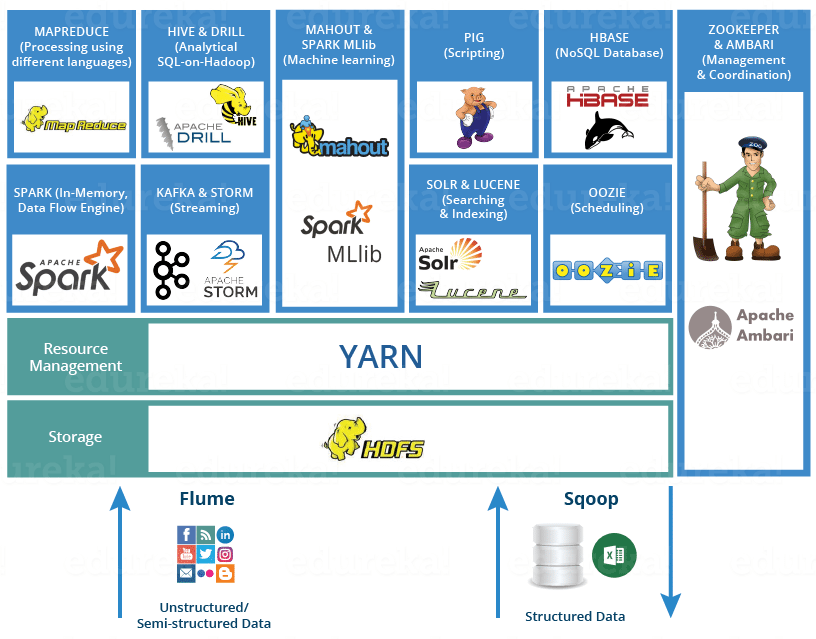
In your computer the resources are managed by your operating system, right? just like that in hadoop the resources are managed by YARN.

HADOOP ECOSYSTEM

The Hadoop ecosystem is a collection of open-source software projects that are related to Hadoop. These projects provide additional features and functionality that can be used to enhance the capabilities of Hadoop

It is important to remember that The Hadoop ecosystem is constantly evolving, and new projects are being added all the time.

<https://hadoopecosystemtable.github.io/>



HDP Sandbox

HDP Sandbox is a Hadoop sandbox environment. What is a sandbox environment though?

A sandbox environment is an isolated testing environment that enables users to run programs or open files without affecting the original application, system, or platform

It is a preconfigured environment that allows you to quickly try out Hadoop and its components without having to install and configure everything yourself. HDP Sandbox is a great way to learn about Hadoop, develop Hadoop applications, and test Hadoop deployments.

Here are some of the features of HDP Sandbox:

It includes a single-node Hadoop cluster with all of the most popular Hadoop components, including HDFS, YARN, MapReduce, Hive, and Pig.

It is preconfigured with a sample dataset, so you can start exploring Hadoop right away.

It is easy to install and use.

It is available for free.

If you are interested in learning about Hadoop, I recommend checking out HDP Sandbox. It is a great way to get started with Hadoop and see what it can do.