

Block

2

ETL, OLAP AND TRENDS

UNIT 4

Extract, Transform and Loading

57

UNIT 5

Introduction to Online Analytical Processing

71

UNIT 6

Trends in Data Warehouse

89

PROGRAMME DESIGN COMMITTEE

Prof. (Retd.) S.K. Gupta , IIT, Delhi

Prof. T.V. Vijay Kumar JNU, New Delhi

Prof. Ela Kumar, IGDTUW, Delhi

Prof. Gayatri Dhingra, GVMITM, Sonipat

Mr. Milind Mahajan,, Impressico Business Solutions, New Delhi

Sh. Shashi Bhushan Sharma, Associate Professor, SOCIS, IGNOU

Sh. Akshay Kumar, Associate Professor, SOCIS, IGNOU

Dr. P. Venkata Suresh, Associate Professor, SOCIS, IGNOU

Dr. V.V. Subrahmanyam, Associate Professor, SOCIS, IGNOU

Sh. M.P. Mishra, Assistant Professor, SOCIS, IGNOU

Dr. Sudhansh Sharma, Assistant Professor, SOCIS, IGNOU

COURSE DESIGN COMMITTEE

Prof. T.V. Vijay Kumar, JNU, New Delhi

Dr. Rahul Johri, USICT, GGSIPU, New Delhi

Mr. Vinay Kumar Sharma, NVLI, IGNOU

Sh. Shashi Bhushan Sharma, Associate Professor, SOCIS, IGNOU

Sh. Akshay Kumar, Associate Professor, SOCIS, IGNOU

Dr. P. Venkata Suresh, Associate Professor, SOCIS, IGNOU

Dr. V.V. Subrahmanyam, Associate Professor, SOCIS, IGNOU

Sh. M.P. Mishra, Assistant Professor, SOCIS, IGNOU

Dr. Sudhansh Sharma, Assistant Professor, SOCIS, IGNOU

SOCIS FACULTY

Prof. P. Venkata Suresh, Director, SOCIS, IGNOU

Prof. V.V. Subrahmanyam, SOCIS, IGNOU

Dr. Akshay Kumar, Associate Professor, SOCIS, IGNOU

Dr. Naveen Kumar, Associate Professor, SOCIS, IGNOU (on EOL)

Dr. M.P. Mishra, Associate Professor, SOCIS, IGNOU

Dr. Sudhansh Sharma, Assistant Professor, SOCIS, IGNOU

Dr. Manish Kumar, Assistant Professor, SOCIS, IGNOU

BLOCK PREPARATION TEAM

Course Editor

Prof. Devendra Kumar Tayal
Dept. of Computer Science & Engineering
Indira Gandhi Delhi Technical University for Women
New Delhi

Language Editor

Prof. Parmod Kumar
School of Humanities
IGNOU
New Delhi

Course Writers

Unit 4: Prof. K. Swathi
NRI Institute of Technology Vijayawada

Unit 5: Prof. Archana Singh
Dept. Of Information Technology
Amity School of Engineering & Technology
Noida

Unit 6: Dr. Archana Rajendra Kachh,
Asst. Professor, University of Mumbai
Mumbai

Course Coordinator: Prof. V.V. Subrahmanyam

Print Production

Mr. Sanjay Aggarwal, Assistant Registrar (Publication), MPDD

April 2023

©Indira Gandhi National Open University, 2022

ISBN- 978-93-5568-774-6

All rights reserved. No part of this work may be reproduced in any form, by mimeograph or any other means, without permission in writing from the Indira Gandhi National Open University.

Further information on the Indira Gandhi National Open University courses may be obtained from the University's office at Maidan Garhi, New Delhi-110068.

Printed and published on behalf of the Indira Gandhi National Open University, New Delhi by MPDD, IGNOU.

Laser Typeset by Raj Printers, A-9, Sector B-2, Tronica City, Loni (Gzb.)

printed at : Rohan Pragya Printing And Packaging Pvt. Ltd. H-76 Site-V, UPSIDC, Kasna

BLOCK INTRODUCTION

The title of the block is ETL, OLAP and Trends. The objectives of this block are to make you understand about the underlying concepts of ETL, OLAP and Trends in Data Warehousing.

The block is organized into 3 units:

Unit 4 covers the overview of extract, transform and loading components of data warehousing;

Unit 5 covers the OLAP, its characteristics, applications and types of OLAP architectures; and

Unit 6 covers the trends like data lakes, cloud data warehousing, real-time data warehousing and data warehouse automation.



UNIT 4 EXTRACT, TRANSFORM AND LOADING

- 4.0 Introduction
- 4.1 Objectives
- 4.2 ETL and its Need
 - 4.2.1 Why do You Need ETL?
- 4.3 ETL Process
 - 4.3.1 Data Extraction
 - 4.3.2 Data Transformation
 - 4.3.3 Data Loading
 - 4.3.3.1 Types of Incremental Loads
 - 4.3.3.2 Challenges in Incremental Loading
- 4.4 Working of ETL
 - 4.4.1 Layered Implementation of ETL in a Data Warehouse
- 4.5 ETL and OLAP Data Warehouses
- 4.6 ETL Tools and their Benefits
- 4.7 Improving the Performance of ETL
- 4.8 ELT and its Need
 - 4.8.1 Why do you Need ELT?
 - 4.8.2 Benefits of ELT
 - 4.8.3 ETL Vs ELT
- 4.9 Summary
- 4.10 Solutions / Answers
- 4.11 Further Readings

4.0 INTRODUCTION

A data warehouse is a digital storage system that connects and harmonizes large amounts of data from many different sources. Data warehouses store current and historical data in one place and act as the single source for an organization. A typical data warehouse has four main components namely:

Central database: A database serves as the foundation of your data warehouse. Traditionally, these have been standard relational databases running on premise or in the cloud. But because of Big Data, the need for true, real-time performance, and a drastic reduction in the cost of RAM, in-memory databases are rapidly gaining in popularity.

Data integration: Data is pulled from source systems and modified to align the information for rapid analytical consumption using a variety of data integration approaches such as ETL (extract, transform, load) and ELT as well as real-time data replication, bulk-load processing, data transformation, and data quality and enrichment services.

Metadata: Metadata is data about your data. It specifies the source, usage, values, and other features of the data sets in your data warehouse. There is business metadata, which adds context to your data, and technical metadata, which describes how to access data – including where it resides and how it is structured.

Data warehouse access tools: Access tools allow users to interact with the data in your data warehouse. Examples of access tools include: query and reporting tools, application development tools, data mining tools, and OLAP tools.

All these components are engineered for speed so that you can get results quickly and analyze the data within no time.

In this unit, we will study about Data Integration component approach such as Extract, Transform and Load (ETL) in detail.

4.1 OBJECTIVES

After going through this unit, you shall be able to:

- understand the purpose ETL;
- describe the ETL process, benefits and ETL tools;
- know the complete working of the ETL;
- discuss various layers involved in the ETL implementation;
- to summarize the functionality of ETL, its need and benefits, and
- to compare and contrast the ETL with ELT.

4.2 ETL AND ITS NEED

Extract, Transform, Load (ETL) as shown in Figure 1, is a process of data integration that encompasses three steps - extraction, transformation, and loading. In a nutshell, ETL systems take large volumes of raw data from multiple sources, convert it for analysis, and load that data into your warehouse.

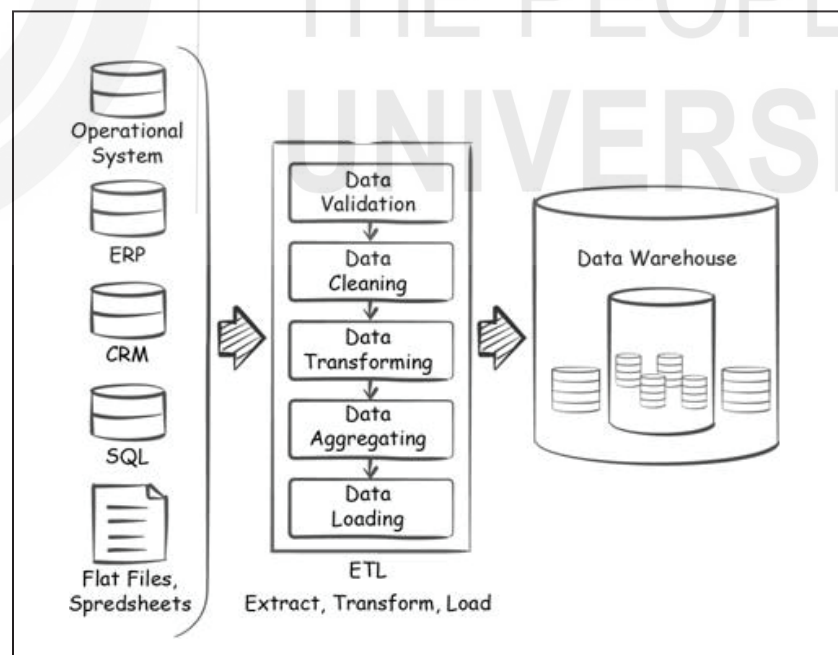


Figure 1: ETL in a Data Warehouse

4.2.1 Why Do You Need ETL?

ETL saves you significant time on data extraction and preparation - time that you can better spend on evaluating your business. Practicing ETL is also part of a healthy

data management workflow, ensuring high data quality, availability, and reliability. Each of the three major components in the ETL saves time and development effort by running just once in a dedicated data flow:

Extract: In ETL, the first link determines the strength of the chain. The extract stage determines which data sources to use, the refresh rate (velocity) of each source, and the priorities (extract order) between them — all of which heavily impact your time to insight.

Transform: After extraction, the transformation process brings clarity and order to the initial data swamp. Dates and times combine into a single format and strings parse down into their true underlying meanings. Location data convert to coordinates, zip codes, or cities/countries. The transform step also sums up, rounds, and averages measures, and it deletes useless data and errors or discards them for later inspection. It can also mask personally identifiable information (PII) to comply with GDPR, CCPA, and other privacy requirements.

Load: In the last phase, much as in the first, ETL determines targets and refresh rates. The load phase also determines whether loading will happen incrementally, or if it will require “upsert” (updating existing data and inserting new data) for the new batches of data.

Let us learn the whole process in the following section.

4.3 ETL PROCESS

ETL collects and processes data from various sources into a single data store (a data warehouse or data lake), making it much easier to analyze. The three steps in ETL process are mentioned below:

4.3.1 Data Extraction

Data extraction involves the following four steps:

Identify the data to extract: The first step of data extraction is to identify the data sources you want to incorporate into your data warehouse. These

sources might be from relational SQL databases like MySQL or non-relational NoSQL databases like MongoDB or Cassandra. The information could also be from a SaaS platform like Salesforce or other applications. After identifying the data sources, you need to determine the specific data fields you want to extract.

Estimate how large the data extraction is: The size of the data extraction matters. Are you extracting 50 megabytes, 50 gigabytes, or 50 petabytes of data? A larger quantity of data will require a different ETL strategy. For example, you can make a larger dataset more manageable by aggregating it to month-level rather than day-level, which reduces the size of the extraction. Alternatively, you can upgrade your hardware to handle the larger dataset.

Choose the extraction method: Since data warehouses need to update continually for the most accurate reports, data extraction is an ongoing process that may need to happen on a minute-by-minute basis. There are three principal methods for extracting information:

- (a) **Update notifications:** The preferred method of extraction involves update notifications. The source system will send a notification when one of its

records has changed, and then the data warehouse updates with only the new information.

- (b) **Incremental extraction:** The second method, which you can use when update notifications aren't possible, is incremental extraction. This involves identifying which records have changed and performing extraction of only those records. A potential setback is that incremental extraction cannot always identify deleted records.
- (c) **Full extraction:** When the first two methods won't work, a complete update of all the data through full extraction is necessary. Keep in mind that this method is likely only feasible for smaller data sets.

Assess your SaaS platforms: Businesses formerly relied on in-house applications for accounting and other record-keeping. These applications used OLTP transactional databases that they maintained on an on-site server. Today, more businesses use SaaS (software as a service) platforms like Google

Analytics, HubSpot, and Salesforce. To pull data from one of these, you'll need a solution that integrates with the unique API of the platform. Xplenty is one such solution.

4.3.2 Data Transformation

In traditional ETL strategies, data transformation that occurs in a staging area (after extraction) is "multistage data transformation". In ELT, data transformation that happens after loading data into the data warehouse is "in-warehouse data transformation". You may need to perform some of the following data transformations:

Deduplication (normalizing): Identifies and removes duplicate information.

Key restructuring: Draws key connections from one table to another.

Cleansing: Involves deleting old, incomplete, and duplicate data to maximize data accuracy - perhaps through parsing to remove syntax errors, typos, and fragments of records.

Format revision: Converts formats in different datasets - like date/time, male/female, and units of measurement - into one consistent format.

Derivation: Creates transformation rules that apply to the data. For example, maybe you need to subtract certain costs or tax liabilities from business revenue figures before analyzing them.

Aggregation: Gathers and searches data so you can present it in a summarized report format.

Integration: Reconciles diverse names/values that apply to the same data elements across the data warehouse so that each element has a standard name and definition.

Filtering: Selects specific columns, rows, and fields within a dataset.

Splitting: Splits one column into more than one column.

Joining: Links data from two or more sources, such as adding spend information across multiple SaaS platforms.

Summarization: Creates different business metrics by calculating value totals. For

example, you might add up all the sales made by a specific salesperson to create total sales metrics for specific periods.

Validation: Sets up automated rules to follow in different circumstances. For instance, if the first five fields in a row are NULL, then you can flag the row for investigation or prevent it from being processed with the rest of the information.

4.3.3 Data Loading

Data loading is the process of loading the extracted information into your target data repository. Loading is an ongoing process that could happen through “full loading” (the first time you load data into the warehouse) or “incremental loading” (as you update the data warehouse with new information). Because incremental loads are the most complex, we'll focus on them in this section.

4.3.3.1 Types of Incremental Loads

Incremental loads extract and load information that has appeared since the last incremental load. This can happen in two ways: (a) Batch incremental loads and (b) Streaming incremental loads.

- (a) **Batch incremental loads:** The data warehouse ingests information in packets or batches. If it's a large batch, it's best to carry out a batch load during off-peak hours - on a daily, weekly, or monthly basis - to prevent system slowdowns. However, modern data warehouses can also ingest small batches of information on a minute-by-minute basis with an ETL platform like Xplenty. This allows them to achieve an approximation of real-time updates for the end-user.
- (b) **Streaming incremental loads:** The data warehouse ingests new data as it appears in real-time. This method is particularly valuable when the end-user requires real-time updates (for example: for up-to-the-minute decision-making). Further, streaming incremental loads are only possible when the updates involve a very small amount of data. In most cases, minute-by-minute batch updates offer a more robust solution than real-time streaming.

4.3.3.2 Challenges in Incremental Loading

Incremental loads can disrupt system performance and cause a host of problems, including:

Data structure changes: Data formats in your data sources or data warehouse may need to evolve according to the needs of your information system. However, changing one part of the system could lead to incompatibilities that interfere with the loading process. To prevent problems relating to inconsistent, corrupt, or incongruent data, it's important to zoom out and review how slight changes affect the total ecosystem before making the appropriate adjustments.

Processing data in the wrong order: Data pipelines can follow complex trajectories that result in your data warehouse processing, updating, or deleting information in the wrong order. That can lead to corrupt or inaccurate information. For this reason, it's vital to monitor and audit the ordering of data processing.

Failure to detect problems: Quick detection of any problems with your ETL workflow is crucial: e.g. when an API goes down, when your API access credentials are out-of-date, when system slowdowns interrupt dataflow from an API or when the target data warehouse is down. The sooner you detect the problem, the faster

you can fix it, and the easier it is to correct the inaccurate/corrupt data that results from it.

4.4 WORKING OF ETL

In this section, we'll dive a little deeper, taking an in-depth look at each of the three steps in the ETL process.

You can use scripts to implement ETL (i.e. custom do it yourself code) or you can use a dedicated ETL tool. An ETL system performs a number of important functions, including:

- (a) **Parsing/Cleansing:** Data generated by applications may be in various formats like JSON, XML, or CSV. The parsing stage maps data into a table format with headers, columns, and rows, and then extracts specified fields.
- (b) **Data Enrichment:** Preparing data for analytics usually requires certain data enrichment steps, including injecting expert knowledge, resolving discrepancies, and correcting bugs.
- (c) **Setting Velocity:** “Velocity” refers to the frequency of data loading, i.e. inserting new data and updating existing data.
- (d) **Data Validation:** In some cases, data is empty, corrupted, or missing crucial elements. During data validation, ETL finds these occurrences and determines whether to stop the entire process, skip the data or set the data aside for human inspection.

4.4.1 Layered Implementation of ETL in a Data Warehouse

When an ETL process is used to move data into a data warehouse, a separate layer represents each phase:

- (a) **Mirror/Raw layer:** This layer is a copy of the source files or tables, with no logic or enrichment. The process copies and adds source data to the target mirror tables, which then hold historical raw data that is ready to be transformed.
- (b) **Staging layer:** Once the raw data from the mirror tables transform, all transformations wind up in staging tables. These tables hold the final form of the data for the incremental part of the ETL cycle in progress.
- (c) **Schema layer:** These are the destination tables, which contain all the data in its final form after cleansing, enrichment, and transformation.
- (d) **Aggregating layer:** In some cases, it's beneficial to aggregate data to a daily or store level from the full dataset. This can improve report performance, enable the addition of business logic to calculate measures, and make it easier for report developers to understand the data.

4.5 ETL AND OLAP DATA WAREHOUSES

Data engineers have been using ETL for over two decades to integrate diverse types of data into online analytical processing (OLAP) data warehouses. The reason for doing this is simple: to make data analysis easier. Normally, business applications use online transactional processing (OLTP) database systems. These are optimized for writing, updating, and editing the information inside them. They are not good at reading and analysis. However, online analytical processing database systems

are excellent at high-speed reading and analysis. That's why ETL is necessary to transform OLTP information, so it can work with an OLAP data warehouse.

During the ETL process, information is:

- i. Extracted from various relational database systems (OLTP or RDBMS) and other sources.
- ii. Transformed within a staging area, into a compatible relational format, and integrated with other data sources.
- iii. Loaded into the online analytical processing (OLAP) data warehouse server.

In the past, data engineers hand-coded ETL pipelines in R, Python, and SQL - a laborious process that could take months to complete. Today, hand-coded ETL continues to be necessary in many cases. However, modern ETL solutions like Xplenty allow data teams to skip hand-coding and automatically integrate the most popular data sources into their data warehouses. This has dramatically increased the speed of setting up an ETL pipeline, while eliminating the risk of human error.

4.6 ETL TOOLS AND THEIR BENEFITS

ETL tools come in a wide variety both in open source or proprietary categories. There are ETL frameworks and libraries that you can use to build ETL pipelines in Python. There are tools and frameworks you can leverage for GO and Hadoop. Really, there is an open-source ETL tool out there for almost any unique ETL need. The downside, of course, is that you'll need lots of custom coding, setup, and man-hours getting the ETL operational. Even then, you may find that you need to tweak your ETL stack whenever you introduce additional tasks. Following are some of the benefits of the ETL tools:

- **Scalability:** Trying to scale-out hand-coded ETL solutions is difficult. As schema complexity rises and your tasks grow more complex and resource-hungry, establishing solid pipelines and deploying the necessary ETL resources can become impossible. With cloud-based ETL tools like Xplenty, you have unlimited scalability at the click of a button.
- **Simplicity:** Going from a hand-coded ETL solution using SQLAlchemy and pandas with rpy2 and parse to something as simple as a cloud-based ETL can be life changing. The benefits of having all of your needs layered into one tool save you time, resources, and lots of headaches.
- **Out-of-the-box:** While open source ETL tools like Apache Airflow require some customization, cloud-based ETL tools like Xplenty work out-of-the-box.
- **Compliance:** The overwhelming nature of modern data compliance can be frightening. Between GDPR, CCPA, HIPAA, and all of the other compliance and privacy nets, using an ETL tool that bakes compliance into its framework is an easy way to skip difficult and risky compliance setups.
- **Long-term costs:** Hand-coded solutions may be cheaper up-front, but they will cost you in the long run. The same thing could be said about open source ETL tools. Since you have to spend time and energy on modification, you're forced to onboard early or risk delaying project launches. Cloud-based ETL tools handle maintenance and back-end caretaking for you.

4.7 IMPROVING THE PERFORMANCE OF ETL

Ultimately tuning is very much required for the ETL to perform better. Following are some of the factors to be considered to improve the ETL performance:

(i) Tackle the Bottlenecks

Before anything else, make sure you log metrics such as time, the number of records processed, and hardware usage. Check how many resources each part of the process takes and address the heaviest one. Usually, it will be the second part, building facts, and dimensions in the staging environment.

(ii) Load Data Incrementally

Loading only the changes between the previous and the new data saves a lot of time as compared to a full load. It's more difficult to implement and maintain, but difficult doesn't mean impossible, so do consider it. Loading incrementally can definitely improve the ETL performance.

(iii) Partition Large Tables

If you use relational databases and you want to improve the data processing window, you can partition large tables. That is, cut big tables down to physically smaller ones, probably by date. Each partition has its own indices and the indices tree is shallower thus allowing for quicker access to the data. It also allows switching data in and out of a table in a quick metadata operation instead of actual insertion or deletion of data records.

(iv) Cut Out Extraneous Data

It's important to collect as much data as possible, but not all of it is worthy enough to enter the data warehouse. To improve the ETL performance, define exactly which data should be processed and leave irrelevant rows/columns out. Better to start small and grow as you go as opposed to creating a giant data that takes much time to process.

(v) Cache the Data

Caching data can greatly speed things up since memory access performs faster than do hard drives. Note that caching is limited by the maximum amount of memory your hardware supports.

(vi) Process in Parallel using Hadoop

Instead of processing serially, optimize resources by processing in parallel.

Sort and aggregate functions (count, sum, etc.) block processing because they must end before the next task can begin. Even if you can process in parallel, it won't help if the machine is running on 100% CPU the entire time. You could scale up by upgrading the CPU, but it would scale only to a limit. Hadoop is a much better solution.

Apache Hadoop is designed for the distributed processing of large data over a cluster of machines. It uses HDFS, a dedicated file system that cuts data into small chunks and optimally spreads them over the cluster. Duplicate copies are kept and the system maintains integrity automatically.

MapReduce is used to process tasks (Hadoop 2 or YARN allows more applications). Each MapReduce job works in 2 stages:

- (a) Map - filtering and sorting data - tasks are divided into sub-tasks and processed in parallel by the cluster machines.
- (b) Reduce - summary operations - data from the previous stage is combined.

Hadoop is optimized for distributed processing analytics. Sort and aggregate functions execute in parallel on an entire cluster.

4.8 ELT AND ITS NEED

Extract/load/transform (ELT) is the process of extracting data from one or multiple sources and loading it into a target data warehouse. Instead of transforming the data before it's written, ELT takes advantage of the target system to do the data transformation. This approach requires fewer remote sources than other techniques because it needs only raw and unprepared data. The process is illustrated in Figure 2.

ELT is an alternative to the traditional extract/transform/load (ETL) process. It pushes the transformation component of the process to the target database for better performance. This capability is very useful for processing the massive data sets needed for business intelligence (BI) and big data analytics.

Because it takes advantage of the processing capability already built into a data storage infrastructure, ELT reduces the time data spends in transit and boosts efficiency.

It is becoming increasingly common for data to be extracted from its source locations, then loaded into a target data warehouse to be transformed into actionable business intelligence. ELT process consists of three steps:

- a. **Extract** - This step works similarly in both ETL and ELT data management approaches. Raw streams of data from virtual infrastructure, software, and applications are ingested either in their entirety or according to predefined rules.
- b. **Load** – The ELT differs here with the ETL. Rather than deliver this mass of raw data and load it to an interim processing server for transformation, ELT delivers it directly to the target storage location. This shortens the cycle between extraction and delivery.
- c. **Transform** - The database or data warehouse sorts and normalizes the data, keeping part or all of it on hand and accessible for customized reporting. The overhead for storing this much data is higher, but it offers more opportunities to mine it for relevant business intelligence in near real-time.

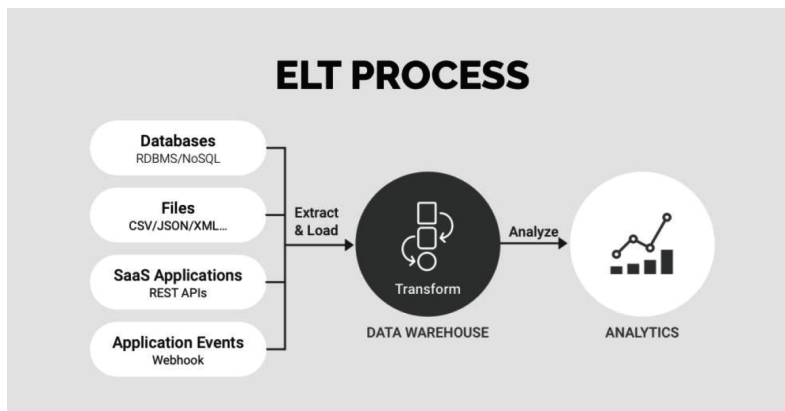


Figure 2: ELT Process

4.8.1 Why Do You Need ELT?

Transforming data after uploading it to modern cloud ecosystems is most effective for:

- Large enterprises with vast data volumes
- Businesses that collect data from multiple source systems or in dissimilar formats
- Companies that require quick or frequent access to integrated data
- Data scientists who rely on business intelligence
- IT departments and data stewards interested in a low-maintenance solution

The ELT process improves data conversion and manipulation capabilities due to parallel load and data transformation functionality. This schema allows data to be accessed and queried in near real time.

However, you might want to stick with ETL if you have dirty data (e.g., duplicate records, incomplete or inaccurate data) that will require data engineers to clean and format it prior to data loading.

4.8.2 Benefits of ELT

With traditional ETL, relevant data is transformed before it is uploaded to a data warehouse, and then it must be pushed out of the warehouse for analysis or processing. This data pipeline works, but it can take more time to migrate data from the source to the target system.

The ELT process saves you steps and time. Data is first loaded into the target ecosystem, such as a data warehouse, and then transformed. Authorized users can securely access the data without returning it to source systems. No downloading is necessary for it. There are reasons to continue using ETL tools. For example, some companies want to keep all their data on-premises. If there is a small amount of data, and it is relational and structured, traditional ETL is effective for businesses that favor hands-on data integration. However, the ELT approach has several benefits for most industries which are listed below:

a) Get better results with more efficient effort

ELT allows you to integrate and process large amounts of data, both structured and unstructured from multiple servers. And, both raw and cleansed data can be accessed with artificial intelligence (AI) and machine learning (ML) tools in addition to SQL and NoSQL processing.

b) Transform your data faster

ELT doesn't have to wait for the data to be transformed and then loaded. The transformation process happens where the data resides, so you can access your data in a few seconds, a huge benefit when processing time-sensitive data.

c) Combine data from different sources and formats

Larger enterprises typically have multiple, disparate data sources like onsite servers, cloud warehouses and log files. Using ELT means you can combine data from various data sets regardless of the source or whether it is structured or unstructured, related or unrelated.

d) Manage data at scale

Technological advances allow organizations to collect petabytes (a million gigabytes!) of data. ELT streamlines the management of massive amounts of data by allowing raw and cleansed data to be stored and accessed. If you're planning to use cloud-based data

warehousing or high-end data processing engines like Hadoop, ELT can take advantage of the native processing power for greater scalability.

e) Save time and money

ELT reduces the time data spends in transit and doesn't require an interim data system or additional remote resources to transform the data outside the cloud. Plus, there's no need to move data in and out of cloud ecosystems for analysis. The more your data moves around, the more the costs add up. The scalability of ELT makes it cost-effective for businesses of any size.

4.8.3 ETL Vs ELT

The primary differences between ETL and ELT are how much data is retained in data warehouses and where data is transformed. With ETL, the transformation of data is done before it is loaded into a data warehouse. This enables analysts and business users to get the data they need faster, without building complex transformations or persistent tables in their business intelligence tools. Using the ELT approach, data is loaded into the warehouse as is, with no transformation before loading. This makes jobs easier to configure because it only requires an origin and a destination.

The ETL and ELT approaches to data integration differ in several key ways as listed below:

- **Load time** - It takes significantly longer to get data from source systems to the target system with ETL.
- **Transformation time** - ELT performs data transformation on-demand, using the target system's computing power, reducing wait times for transformation.
- **Complexity** - ETL tools typically have an easy-to-use GUI that simplifies the process. ELT requires in-depth knowledge of BI tools, masses of raw data, and a database that can transform it effectively.
- **Data warehouse support** - ETL is a better fit for legacy on-premise data warehouses and structured data. ELT is designed for the scalability of the cloud.
- **Maintenance** - ETL requires significant maintenance for updating data in the data warehouse. With ELT, data is always available in near real-time.

Both ETL and ELT processes have their importance in the data warehouse architecture as per the understanding of business unique needs and strategies which is key to determining which process will deliver the best results.

Check your Progress 1

1. Define Data Extraction process of ETL along the variety of sources of data accounted for this process. Also, mention the challenge(s) for an ETL tool during the extracting process.

.....

2. Describe the Data Transformation process involved in the ETL.

.....

3. Discuss briefly the Data Loading process of ETL.

.....

4.9 SUMMARY

The process of extracting data from source systems and bringing it into the data warehouse is commonly called ETL process, which stands for extraction, transformation, and loading. In this unit, we had studied that ETL refers to a broad process. The methodology and tasks of ETL have also been studied.

Apart from the ETL, we had studied a new approach known as ELT and its benefits too, in this unit.

In the next unit, we will be study about Online Analytical Processing (OLAP).

4.10 SOLUTIONS / ANSWERS

Check your Progress 1:

1. Extracting data is the act of pulling data from one or more data sources. During the extraction phase of ETL, you may handle a variety of sources with data, such as:

- Relational and non-relational databases
- Flat files (e.g. XML, JSON, CSV, Microsoft Excel spreadsheets, etc.)
- SaaS applications, such as CRM (customer relationship management) and ERP (enterprise resource planning) systems
- APIs (application programming interfaces)
- Websites
- Analytics and monitoring tools
- System logs and metadata

In the first step, extracted data sets come from a source (e.g., Salesforce, Google AdWords, etc.) into a staging area. The staging area acts as a buffer between the data warehouse and the source data. Since data may be coming from multiple

different sources, it's likely in various formats, and directly transferring the data to the warehouse may result in corrupted data. The staging area is used for data cleansing and organization.

A big challenge during the data extraction process is how your ETL tool handles structured and unstructured data. All of those unstructured items (e.g., emails, web pages, etc.) can be difficult to extract without the right tool, and you may have to create a custom solution to assist you in transferring unstructured data if you chose a tool with poor unstructured data capabilities.

2. It's rarely the case that your extracted data is already in the exact format that you need it to be. For example, it may require the following:
 - Rearrange unstructured data into a structured format.
 - Limit the data you've extracted to just a few fields.
 - Sort the data so that all the columns are in a certain order.
 - Join multiple tables together.
 - Clean the data to eliminate duplicate and out-of-date records.

The data cleaning and organization stage is the transformation stage. All of that data from multiple source systems will be normalized and converted to a single system format — improving data quality and compliance. ETL yields transformed data through these methods:

- Cleaning
 - Filtering
 - Joining
 - Sorting
 - Splitting
 - Deduplication
 - Summarization
3. Once the process has transformed, sorted, cleaned, validated, and prepared the data, you need to load it into data storage somewhere and the most common target database is a data warehouse. Depending upon your business needs, data can be loaded in batches or all at once. The exact nature of the loading will depend upon the data source, ETL tools, and various other factors.

4.11 FURTHER READINGS

1. William H. Inmon, *Building the Data Warehouse*, Wiley, 4th Edition, 2005.
2. *Data Warehousing Fundamentals*, Paulraj Ponnaiah, Wiley Student Edition, 2001.
3. *Data Warehousing*, Reema Thareja, Oxford University Press, 2009.

UNIT 5 INTRODUCTION TO ONLINE ANALYTICAL PROCESSING

Structure

- 5.0 Introduction
- 5.1 Objectives
- 5.2 OLAP and its Need
- 5.3 Characteristics of OLAP
- 5.4 OLAP and Multidimensional Analysis
 - 5.4.1 Multidimensional Logical Data Modeling and its Users
 - 5.4.2 Multidimensional Structure
 - 5.4.3 Multidimensional Operations
- 5.5 OLAP Functions
- 5.6 Data Warehouse and OLAP: Hypercube and Multicubes
- 5.7 Applications of OLAP
- 5.8 Steps in the OLAP Creation Process
- 5.9 Advantages of OLAP
- 5.10 OLAP Architectures - MOLAP, ROLAP, HOLAP, DOLAP
- 5.11 Summary
- 5.12 Solutions/Answers
- 5.13 Further Readings

5.0 INTRODUCTION

In the earlier unit you had studied about Extract, Transform and Loading (ETL) of a Data Warehouse. Within the data science field, there are two types of data processing systems: online analytical processing (OLAP) and online transaction processing (OLTP). The main difference is that one uses data to gain valuable insights, while the other is purely operational. However, there are meaningful ways to use both systems to solve data problems. OLAP is a system for performing multi-dimensional analysis at high speeds on large volumes of data. Typically, this data is from a data warehouse, data mart or some other centralized data store. OLAP is ideal for data mining, business intelligence and complex analytical calculations, as well as business reporting functions like financial analysis, budgeting and sales forecasting.

In this unit we will focus on Online Analytical Processing (OLAP).

5.1 OBJECTIVES

After going through this unit, you should be able to:

- understand the purpose of a OLAP;
- describe the motivation and benefits of OLAP;
- discuss Multidimensional Modeling;
- describe various OLAP operations;

- list multi cube Applications and steps to create OLAP server, and
- discuss between various types of OLAP like MOLAP, ROLAP, DOLAP and HOLAP.

5.2 OLAP AND ITS NEED

Online Analytical Processing (OLAP) is the technology to analyze and process data from multiple sources at the same time. It accesses the multiple databases at the same time. It is a software which helps the data analysts to collect data from different perspective for developing effective business strategies. The query operations like group, join or aggregation can be easily done with OLAP using pre-calculated or pre-aggregated data hence making it much faster than simple relational databases. You can understand OLAP as a multi cubic structure, which has many cubes, each cube is pertaining to some database. The cubes are designed in such a way that generates reports effectively and efficiently.

OLAP is the core component of the data warehouse implementation, providing fast and flexible multi-dimensional data analysis for business intelligence (BI) and decision support applications. OLAP (for online analytical processing) is a software used to perform high-speed, multivariate analysis of large amounts of data in data warehouses, data markets, or other unified and centralized data warehouses. The data is broken down for display, monitoring or analysis. For example, sales figures can be related to location (region, country, state/province, company), time (year, month, week, day), product (clothing, male/female/child, brand, type), etc., but In a data warehouse, records are stored in tables, and each table can only sort data on two of the dimensions at a time. Recording and reorganizing them into a multi-dimensional format allows very fast processing and very in-depth analysis

The primary objective of OLAP or data analysis is not just data processing .For instance, If a company might compare their sales in the month of January with the month of February then compare those results with another location which may be stored in a separate database. In this case, it needs a multi-view of database design storing all the data categories. Another example of Amazon, it analyzes purchases made by its customers to recommend the customers with a personalized home page of products which are likely to be interested by them. So, this is one of the good examples of OLAP systems. It creates a single platform for all type of business analytical means which includes planning budgeting forecasting and analysis the main benefit of OLAP is the consistency of information and calculations using OLAP systems we can easily apply security restrictions on users and objects to comply with regulations and protect sensitive data.

OLAP assists managers in making decisions by giving multidimensional record views that are efficient to provide, hence enhancing their productivity. Due to the inherent flexibility support provided by organized databases, OLAP functions are self-contained. Through extensive control of analysis-capabilities, it permits simulation of business models and challenges.

Let's see the need to use OLAP to have better understanding of OLAP over relational databases:

- 1) Efficient and Effective methods to improve the sales of an Organization: In retail, having multiple products with different number of channels for selling the product across the globe. OLAP makes it effective and efficient

to search for a product in a different region within a specified time period (like, excluding weekdays sales or just weekend sales or festival duration sales very specific from a very large data distributed.)

- 2) It improves the sales of a business. The data analysis power of OLAP brings effective results in sales. It helps in identifying expenditures which produce a high return of investments (ROI).

Usually, data operations and analysis are performed using the simple spreadsheet, where data values are arranged in row and column format. This is ideal for two-dimensional data. However, OLAP contains multidimensional data, with data usually obtained from a different and unrelated source. Using a spreadsheet is not an optimal option. The cube can store and analyze multidimensional data in a logical and orderly manner.

5.3 CHARACTERISTICS OF OLAP

The main characteristics of OLAP are as follows:

- **Fast:** OLAP acts as a bridge between Data Warehouse and front-end. Hence helps in the better accessibility of data yielding faster results.
- **Analysis:** OLAP data analysis and computational measure and their results are stored in separate data files. OLAP distinguishes better zero and missing values. It should ignore missing value and performs the correct aggregate values. OLAP facilitates interactive query handling and complex analysis for the users.
- **Shared:** OLAP operations drill-down or roll-up, it navigates between various dimensions in multidimensional cube making it effective and efficient reporting system.
- **Multidimensional:** OLAP has Multidimensional conceptual view and access of data to different users at different levels. The increasing number of dimensions and report generation performance of the OLAP system does not significantly degrade.
- **Data and Information:** OLAP has calculation power for complex queries and data. It does data visualization using graphs and charts.

5.4 OLAP AND MULTIDIMENSIONAL ANALYSIS

The multi-dimensional data model stores data in the form of data cube. In a data warehouse. Generally, it supports two- or three-dimension cubes. It gives the data different views and perspectives. Practically in retail store the data is maintained month wise, item wise, region wise thus involving many different dimensions.

5.4.1 Multidimensional Logical Data Modeling and its Users

The multidimensional data modeling provides:

- Different views and perspectives to the data from different angles. The business users have a dimensional and logical view of the data in the data warehouse.
- Multidimensional conceptual view: It allows users to have a dimensional and logical view of the data.

- Multidimensional modeling creates environment for multiuser. Since the OLAP techniques are shared, the OLAP and database operations, containing retrieval, update, adequacy control, integrity, and security can be easily performed.

For example, in the Figure 1, it is shown that the dimensions Time, Regions and Products of a company can be logically saved in a cube. In Figure 2, in the cross tabular form in every quarter, products quantity are shown. In Figure 1, Products, Time and Regions these dimensions can be combined into cubes you can imagine what two dimensions would look like by using a spreadsheet metaphor with the time dimension as the columns and the products dimension as the rows if we add data to this view such as units sold that would be a measure. Measures can be any quantity such as revenue / expenses / unit's / statistics or any text or numerical value if we consider adding the third dimension regions then you can imagine each region being represented as an additional spreadsheet this is how it works when you're limited to a two-dimensional spreadsheet. however, an OLAP cube can represent all three dimensions as a single data set which allows users to fluidly explore all the data from any perspective and despite its name a cube can hold many more than three dimensions so what's the value of using all that to illustrate this.

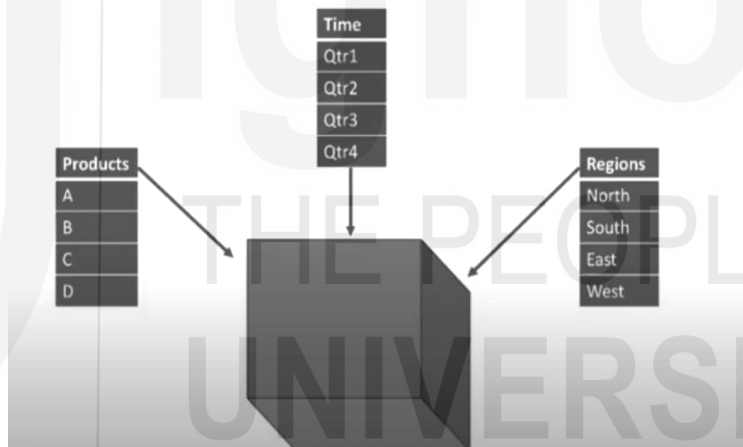


Figure 1: Cube Representation

Products	Time				Regions
	Qtr1	Qtr2	Qtr3	Qtr4	
A	300	200	100	600	
B	500	200	300	100	
C	100	400	100	300	
E	300	400	700	200	

Figure 2: Measurable Data Shown

Let's say that a manager is tracking sales units with three different spreadsheets with three different dimensions products quarters and regions from looking at these spreadsheets. it appears that everything is equal as the manager of these stores would

probably stock them with the same number of items for each product quarter and region. The manager of a store house makes very different decisions to generate a report with just one or two dimensions or by adding more dimensions and reveal more detail which would allow to make better decisions on managing the inventory of the stores. Hence, you can view OLAP facilitates Business Oriented multidimensional data having lot of calculations. The data saved in multidimensional structure is very significant in speed thought analysis to companies to take better decisions. OLAP provides the flexibility of data retrieval to generate reports.

5.4.2 Multidimensional Structure

In the multidimensional model, data are organized into multiple dimensions, and each dimension contains multiple levels of abstraction defined by concept hierarchies. This organization provides users with the flexibility to view data from different perspectives. The data has been organized into multiple dimensions and at each level of dimension, contains multiple levels of abstraction defining the concept hierarchy. It provides flexibility to view data from different angles. Likewise, as explained earlier the conceptual hierarchy of a product is:

Department → Category → Subcategory → Brand → Product

It is important to identify the hierarchy from multi-dimensional cube in terms of query. Then we must look at the performance measure or on which attribute or dimension the query is focused on.

5.4.3 Multidimensional Operations

OLAP provides a user-friendly environment for interactive data analysis. A number of OLAP data cube operations exist to materialize different views of data, allowing interactive querying and analysis of the data.

The most popular end user operations on dimensional data are:

- 1) Roll-up
- 2) Drill-down
- 3) Slice and Dice
- 4) Pivot (rotate)

In daily life we come across operations where the manager is interested in knowing the aggregate of data from the concept hierarchy. It can use the concept hierarchy to roll the data up so for instance instead of a daily aggregated data we have monthly aggregate data and quarterly and then annual year. The concept hierarchy of Time dimension be:

Concept hierarchy of Time dimension



So, to perform this operation, we can roll-up and store the result. Also, it can subtotal those aggregated data. So, if the manager is interested in going down the

concept hierarchy or interested in the minute details to find out the driving attribute responsible for the increase or decrease of sales. For this OLAP operation drill down can be performed.

1) Roll-up:

The roll-up operation (also called drill-up or aggregation operation) performs aggregation on a data cube, either by climbing up a concept hierarchy for a dimension or by climbing down a concept hierarchy, i.e. dimension reduction. In the following example given at figure 3, it is shown a multidimensional cube containing the products of a Home appliances home appliances like laptop, furniture, mobile and kitchen appliances. If the manager wants to view the sales of all the products quarterly, the Roll-up operation can be performed on the categories. In this aggregation process, data is category hierarchy moves up from mobile to the Kitchen store. In the roll-up process at least one or more dimensions get reduced like category here.

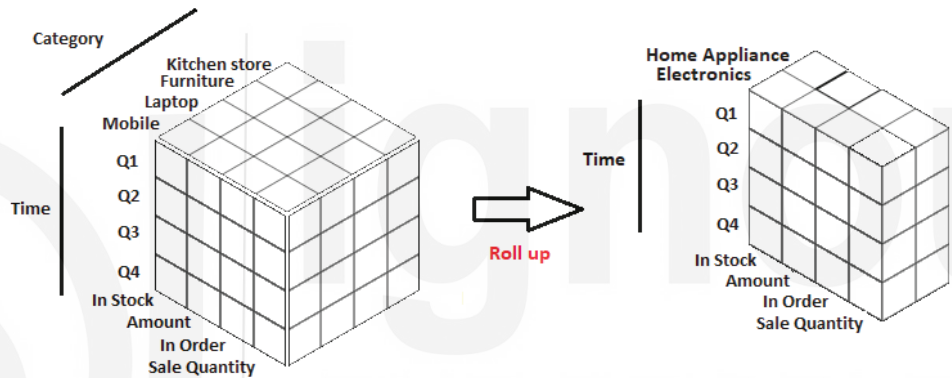


Figure 3: Roll-up on (Category from Home Appliances and Electronics)

It is also known as consolidation. This operation summarizes the data along the dimension.

2) Drill-down:

The drill down operation (also called roll-down) is the reverse of roll up. It navigates from less detailed data to more detailed data. It can be realized by either stepping down a concept hierarchy for a dimension or introducing additional dimensions.

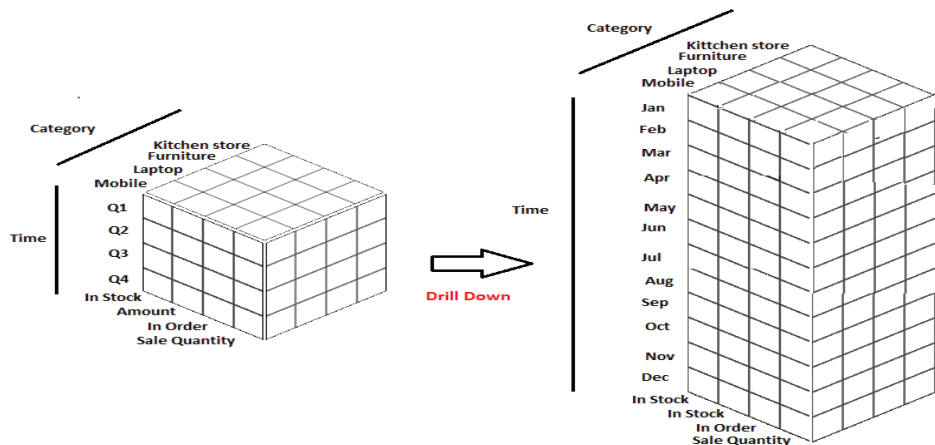


Figure 4: Drill down from Time to Months

You will observe in the above example given at figure 4 a multidimensional cube containing products and time. The Time dimension has been expanded from Quarter → Months to observe the sales month-wise. This is called in Drill down.

3) **Slice:**

This enables an analyst to take one level of information for display. It is another OLAP operation to fetch the data. In this the query on one dimension is triggered in the database and a new sub cube is created.

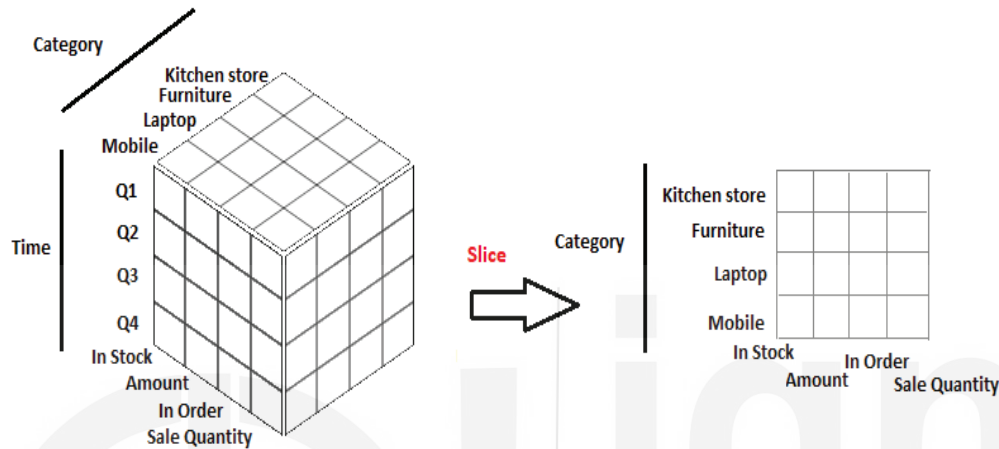


Figure 5: Slice OLAP Operation

In the above figure 5 it can be observed that slice operation is performed on “Time” dimension and a new sub cube is created to retrieve the results.

Slice for Time = “Q1”

4) **Dice:**

This allows an analyst to select data from multiple dimensions to analyze. This OLAP operation is just like the Projection relational query you have read in RDBMS. In this technique you select two or more dimensions that results in the creation of a sub cube as shown in figure 6:

Dice for (Category= “Laptop” or “Mobile”) and (Time = “Q1” or “Q2”) and (Stock = “Amount” or “Sale Quantity”)

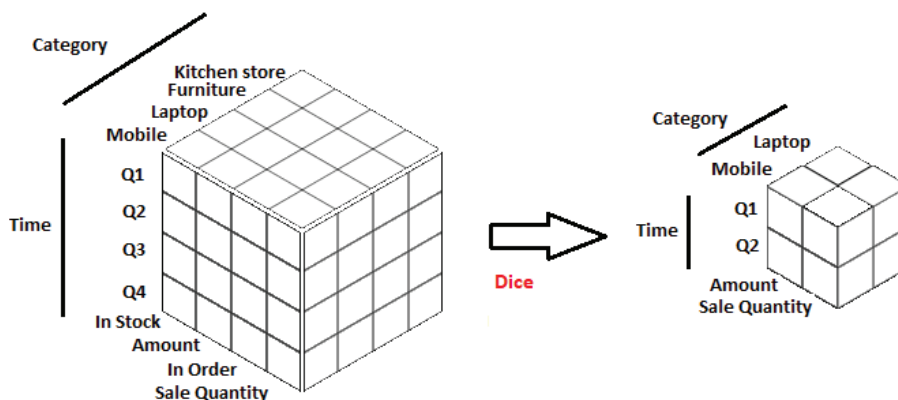


Figure 6: Dice OLAP Operation

4) **Pivot:**

Analysts can gain a new view of data by rotating the data axes of the cube. This OLAP operation fixes one attribute as a Pivot and rotate the cube to fetch the results. Like inverting the spreadsheet it gives a different perspective. You can observe in the figure 7 that the presentation of the dimensions has been changed to impart a different perspective of the data cube for data analysis.

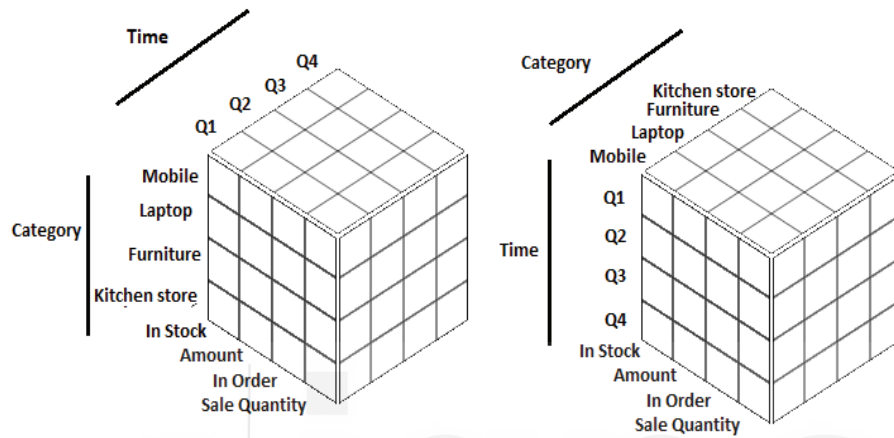


Figure 7: Pivot OLAP Operation



Check Your Progress 1

1) Who are the users of the Multidimensional Data Modeling?

.....

2) What are the five categories of decision support tool?

.....

5.5 OLAP Functions

Online Analytical Processing (OLAP) functions can return the ranking and row numbering. It is very similar to the SQL aggregate functions, however, an aggregate function return an atomic value.

- The OLAP function returns a scalar value of a query. OLAP functions can be performed at the individual row levels too.
- OLAP functions provide data mining functionalities and data analysis. The detailed data analysis and values are supported with OLAP functions.
- The exhaustive and comprehensive data analysis can be achieved row wise unlike simple SQL functions produces results in the form of reports like WITH. OLAP runs on rows of the data warehouse.
- OLAP functions uses SQL commands like INSERT/SELECT/ POPULATE on tables or Views.

5.6 Data Warehouse and OLAP: Hypercube and Multi Cubes

The OLAP cube is a data structure optimized for very quick data analysis. The OLAP Cube consists of numeric facts called measures which are categorized by dimensions. OLAP Cube is also called the hypercube. So, we can say that multidimensional Databases can we see hypercube and multi cube. Multidimensional cubes have smaller multiple cubes and in hypercube it seems there is one cube as logically all the data seems to be as one unit of cube. Hypercube have multiple same dimensions logically. The differences of Multi cube and Hyper cube are shown in Table 1 below:

Table 1: Differences between Multi cube and Hyper cube

	Multi Cube	Hyper Cube
Metadata	Each dimension can belong to many cubes	Each dimension belongs to one cube only
Dimension	Not necessary all the dimensions should belong to some cube	Every dimension owned by a hypercube
Measure Computation	Complex, data can be retrieved from the all the cubes	Simple, as all the numerical facts are available at one place
Multiple	multicube system, if there are two rows in the DIMENSIONS rowset for which the DIMENSION_NAME value is the same (and the CUBE_NAME value is different), these two rows represent the same dimension. As, sub cubes are built from the same pool of available dimensions.	in a multiple hypercube scenario, it is possible for two hypercubes to have a dimension of the same name, each of which has different characteristics. In this case, the DIMENSION_UNIQUE_NAME value is guaranteed to be different.

5.7 APPLICATIONS OF OLAP

OLAP reporting system is widely used in business applications like:

- Sales and Marketing
- Retail Industry
- Financial Organizations – Budgeting
- Agriculture
- People Management
- Process Management

Examples are Essbase from Hyperion Solution and Express Server from Oracle.

☞ Check Your Progress 2

- 1) Explain the OLAP application reporting system in Marketing?

.....

2) What is the purpose of hyper cube. Show slice and dice operation on the sub-cube/hypercube?

.....

3) List the features of an OLAP.

.....

5.8 STEPS IN THE OLAP CREATION

The basic unit of OLAP is an OLAP cube. It is a data structure designed for better and faster retrieval of results from the data analysis. OLAP cubes. It has dimensions with numeric facts. The data arrangement in rows and columns in multidimensional is the logical view not the physical view.

The steps involved in the creation of OLAP are as follows:

Steps to create an OLAP

Step 1: Extract data from variety of sources like text, excel sheets, multimedia files, Online Transaction Processing data in flat files.
Step 2: Transformation and Standardization of data: Since, the data is distributed and incompatible to each other. It involves the data preprocessing or cleaning part where the semantics of databases are changed into a standard form.
Step 3: Loading of data: After all the database nomenclature have been followed then the data is loaded onto the OLAP server or OLAP multidimensional cube.
Step 4: Building of a Cube for data analysis:
<ul style="list-style-type: none"> • Select the dimensions means set of subsets of significant attributes. • Select the concept hierarchies. • Populate the cube with the relevant data • Select the numeric attribute to apply aggregate function.
Step 5: Report Generation

The steps to create OLAP shown in the below figure 8:

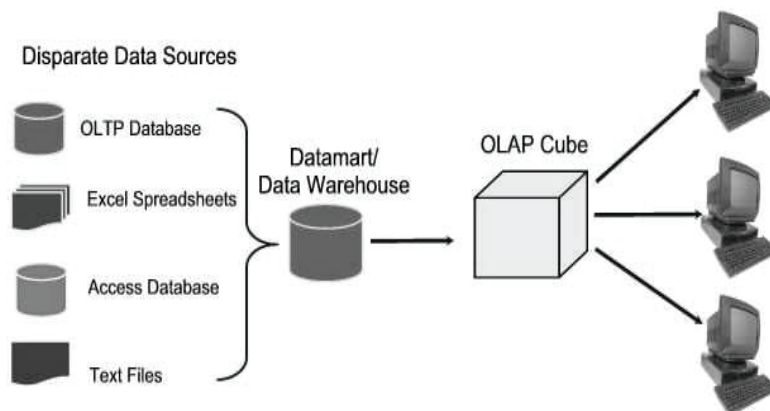


Figure 8 : Steps to create OLAP Cube

5.9 ADVANTAGES OF OLAP

The SQL functions like Group By, Aggregating functions are quite complex to operate in relational databases as compared to multidimensional databases. OLAP can pre-compute the queries can save in sub cubes. The hypercubes also make the computation task faster and saves time. OLAP has proved to an extremely scalable and user – friendly method which is able to perfectly cater to its entire customer needs ranging from small to large companies.

Some listed benefits of using OLAP are as follows:

- ***Data Processing at a faster speed***

The speed of query execution has been tremendous since the use of OLAP technology and is now counted as one of the primary benefits for it. This prevents the customers from spending a lot of time and money on heavy calculations and creating complex reports.

- ***Accessibility***

The cube enables the various kinds of data like – transactional data from various resources, information about every supplier and consumer, etc. all is saved in a concise one location which is easy to operate.

- ***Concise and Fine Data***

OLAP works on the principle of combining multiple and similar records together, which are saved in multiple tables forming a schema between them as a source of connection. These tables combine to form the cube to make the massive information concise and yet finely available to the user. Records can be elemental right down to a single element by “drill down” and back to the cube by “drill up” operations.

- ***Data Representation in Multi-Dimension***

OLAP cube is the center of all the data. Each element of the cube contains various attributes and the number of processes performed on it. The cube axes are outlined by the measure and dimension of the cube which is mostly three - dimensional system. This allows the user to take the information from various slices of the cube. A cube slice is a two – dimensional in nature which gives a clear image of the knowledge trying to be represented.

- ***Business Expressions commonly used***

The size of an OLAP cube consisting of data portrays the company’s economic and financial conditions. The end user does not manipulate the database files; they deal with end processes like products, salesmen, employees, customers, etc. This gives a reason to even user with less to zero technical background to use LAP technology.

- ***Situational Scenarios***

The way the cube can cover almost all parts of a data item is through creating various what – if situations; these what – if situations help in extraction of cube information without tampering the original information on the cube. This feature of OLAP technology is responsible for providing the customers the ability to update the values to look at the consequences brought in the cube’s situation. Through this feature business intelligence can deeply examine the possible factors of driving a situation in a company and prevent them if necessary.

- ***Easily Understood Technology***

Most of the users or customers working on OLAP technology come from a background of less to minimum technology skills. They mostly do not need any unique training to use this technology, which in return helps the company save some money. Moreover, OLAP technology providers provide their end users with enough tutorial, documents and some start off technical assistance particularly in case of web – based OLAP operations. The end customers are given sessions to continuously work with a group of technical experts so that they do not have to solve all the OLAP issues by themselves.

5.10 OLAP ARCHITECTURE: MOLAP, ROLAP, HOLAP AND DOLAP

There are types of OLAP architecture: ROLAP, MOLAP, HOLAP and others as shown in the below figure 9.

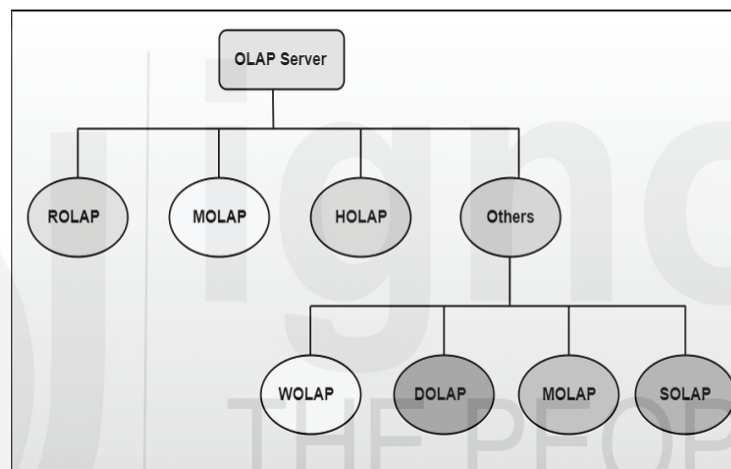


Figure 9: Types of OLAP Architecture

ROLAP Architecture

ROLAP implies Relational OLAP, an application based on relational DBMSs. It performs dynamic multidimensional analysis of data stored in a relational database. The architecture is like three-tiered. It has three components viz. front end (User Interface), ROLAP server (Metadata request processing engine) and the back end (Database Server) as shown in the Figure 10.

- Database server
- ROLAP server
- Front-end tool

In this three-tiered architecture the user submits the request and ROLAP engine converts the request into SQL and submits to the backend database. After the processing of request the engine, it presents the resulting data into multidimensional format to make the task easier for the client to view it.

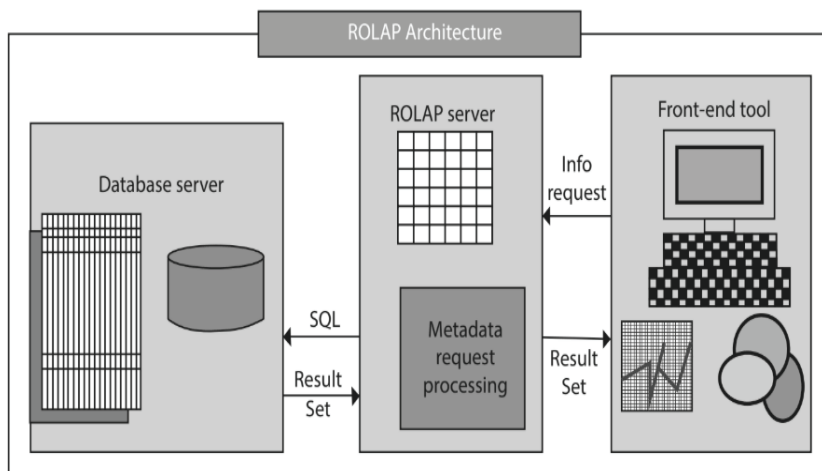


Figure 10 : ROLAP Architecture

The characteristics of ROLAP are:

- ROLAP utilizes the more processing time and disk space.
- ROLAP enables and supports larger user group in the distributed environment.
- ROLAP processes complex queries utilizing the greater amounts of data.

Popular ROLAP products include Metacube by Stanford Technology Group, Red Brick Warehouse by Red Brick Systems.

MOLAP Architecture

MOLAP it stands for Multidimensional Online Analytical Processing. It processes the data using the multidimensional cube using various combinations. Since, the data is stored in multidimensional structure the MOLAP engine uses the pre-computed or pre-stored and stored. The architecture has three components:

- Database server
- MOLAP server
- Front-end tool

MOLAP engine processes pre-compiled information. It has dynamic abilities to perform aggregation of concept hierarchy. MOLAP is very useful in time-series data analysis and economic evaluation. MOLAP in shown in Figure 11.

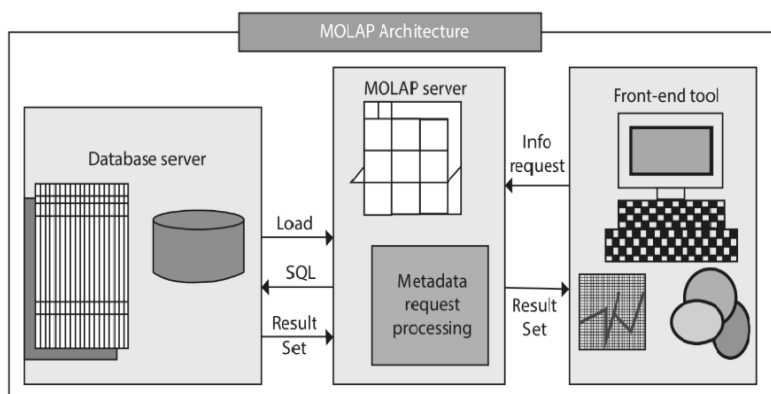


Figure 11 : MOLAP Architecture (Source : internet)

The characteristics of MOLAP are:

- It is a user-friendly architecture, easy to use.
- The OLAP operations slice and dice speeds up the data retrieval.
- It has small pre-computed hypercubes.

Tools that incorporate MOLAP include Oracle Essbase, IBM Cognos, and Apache Kylin.

HOLAP Architecture

It defines Hybrid Online Analytical Processing. It is the hybrid of ROLAP and MOLAP technologies. It connect both the dimensions together in one architecture. It stores the intermediate or part of the data in ROLAP and MOLAP. Depending on the query request it accesses the databases. It stores the relational tables in ROLAP structure, and the data requires multidimensional view are stored and processed using MOLAP architecture as shown in figure 12. It has the following components:

- Database server
- ROLAP and MOLAP server
- Front-end tool

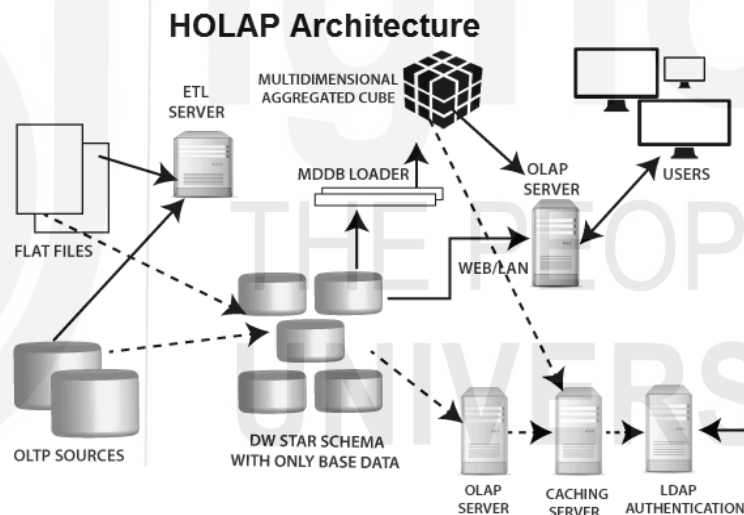


Figure 12 : HOLAP architecture(source: internet)

The characteristics of HOLAP are:

- Flexible handling of data.
- Faster aggregation of data.
- HOLAP can drill down the hierarchy of data and can access to relational database for any relevant and stored information in it.

Popular HOLAP products are Microsoft SQL Server 2000 presents a hybrid OLAP server.

DOLAP Architecture

Desktop Online Analytical Processing (DOLAP) architecture is most suitable for local multidimensional analysis. It is like a miniature of multidimensional database or it's like a sub cube or any business data cube. The components are:

- Database Server
- DOLAP server
- Front End

The characteristics of DOLAP are:

- The three-tiered architecture is designed for low-end, standalone user like a small shop owner in the locality.
- The data cube is locally stored in the system so, retrieval of results is faster.
- No load on the backend or at the server end.
- DOLAP is relatively cheaper to deploy.

☞ **Check Your Progress 3**

- 1) Compare ROLAP, MOLAP and HOLAP.

.....
.....
.....

- 2) Write limitations of OLAP cube.

.....
.....
.....

5.11 SUMMARY

OLAP has proven to be an asset in the field of Business Intelligence as it helps in relieving the large amount of data handling along adding the cost benefits of working with this very technique. Furthermore, OLAP providers normally offer their clients with significant documentation, tutorials, and spark off technical assistance in terms of web-primarily based totally OLAP clients. The customers are continuously loose to deal with the group of tech experts while not having to control all the troubles tied to the software program themselves. The concept hierarchies help to organize the dimensions into logical levels. The various OLAP operations help to extract information across sub cubes. The creation of cube and types of OLAPs helps to understand the architecture and usage of various applications of OLAP.

5.12 SOLUTIONS/ANSWERS

Check Your Progress 1

- 1) Knowledge workers such as data analysts, business analysts, and Executives are the users of OLAP.
- 2) Decision making Tool features are:
 - Report Generation
 - Query Handling
 - EIS (Executive Information System)
 - OLAP (Online Analytical Processing)
 - Data Mining

Check Your Progress 2

- 1) In Marketing, OLAP can be used for various purposes as it helps like planning, budgeting, Financial marketing, sales data analysis and forecasting. The customer experience is very important to all the companies. So, OLAP works very efficiently in analyzing the data of customers, market research analysis, cost-benefit analysis of any project considering all the dimensions.

There are various OLAP tools available. The OLAP tool should have the ability to analyze large amounts of data, data analysis, fast response to the queries and data visualization. For example, IBM Cognos is a very powerful OLAP marketing tool.

- 2) **Purpose of Hypercube in OLAP:** The cube is basically used to represent data with some meaningful measure to compute. Hypercube logically has all the data at one place as a single unit or spreadsheet which makes the computation of queries faster. Each dimension logically belongs to one cube. For example, a multidimensional cube contains data of the cities of India, Product, Sales and Time with conceptual hierarchy (Delhi→2018→Sales). As, shown in below figures.

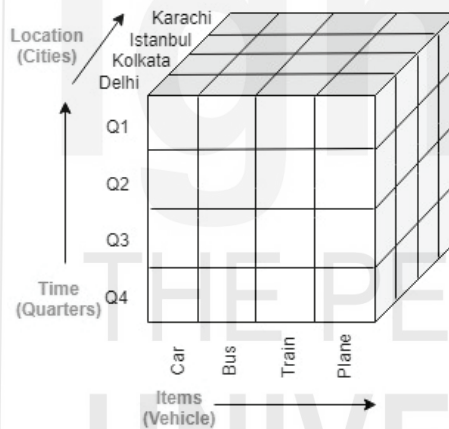


Figure 13: Multidimensional Cube

In the cube given in the overview section, a sub-cube(hypercube) is selected with the following conditions

Location = “Delhi” or “Kolkata” Time = “Q1” or “Q2” , Item = “Car” or “Bus”

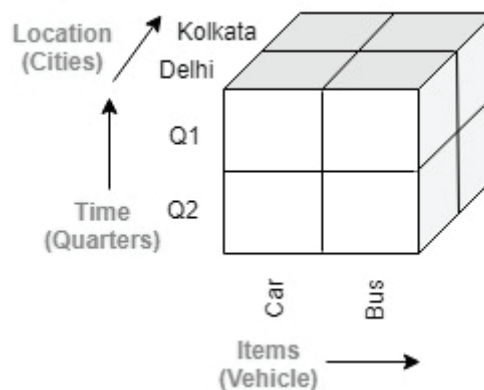


Figure 14 : Hypercube or sub-cube

Slice is performed on the dimension Time = “Q1”.

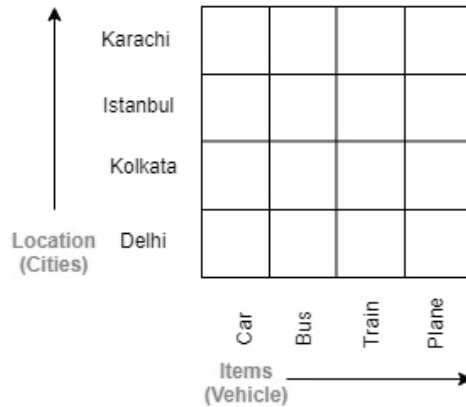


Figure 15 : Slice on Hyper cube

In the sub-cube ,pivot operation is performed.

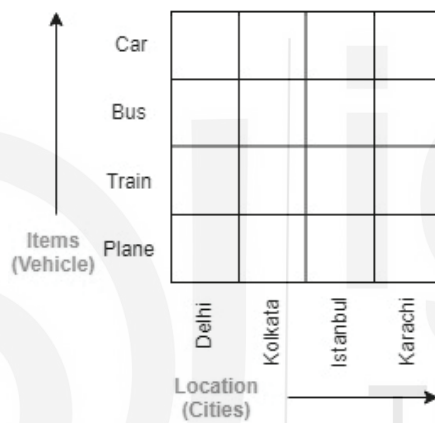


Figure 16: Pivot operation

3) Features of OLAP are:

- Conceptual multidimensional view
- Accessibility of data
- Efficient and flexible Reporting system
- Client/Server architecture
- Supports unrestricted dimensions and aggregation levels
- Uses dynamic sparse matrix handling for faster query results
- Multiuser support

Check Your Progress 3

1) Comparative analysis between ROLAP, MOLAP and HOLAP

Features	ROLAP	MOLAP	HOLAP
Accessibility of data and Processing time	Very slow because of join operation between tables. The data is fetched from data warehouse.	Fast because of multidimensional storage. The data is fetched from multidimensional data cube.	Fast

Features	ROLAP	MOLAP	HOLAP
Storage space requirement	Data is stored in relational tables. Comparatively Large storage space requirement	Data is stored in multidimensional tables. Medium storage space requirements	It uses both ROLAP, MOLAP. Small storage space requirements. No duplicate of data
Latency	Low latency	High latency	Medium latency
Query response time	Slow query response time	Fast query response time.	Medium query response time
Volume of data	Used for large volumes of data	Limited volume of data	Can be used in both scenarios
Retrieval of data	Complex SQL queries are used	Sparse Matrix is used	Both
Data View	Static view of data	Dynamic view of data	Both static and dynamic view of data

2) Limitations of OLAP cube are:

- OLAP requires a star/snowflake schema:
- There is a limited number of dimensions (fields) a single OLAP cube.
- It is nearly impossible to access transactional data in the OLAP cube.
- Changes to an OLAP cube requires a full update of the cube – a lengthy process.

5.13 FURTHER READINGS

- William H. Inmon, Building the Data Warehouse, Wiley, 4th Edition, 2005.
- Data Warehousing Fundamentals, Paulraj Ponnaiah, Wiley Student Edition
- Data Warehousing, Reema Thareja, Oxford University Press
- Data Warehousing, Data Mining & OLAP, Alex Berson and Stephen J.Smith, Tata McGraw – Hill Edition, 2016.

UNIT 6 TRENDS IN DATA WAREHOUSE

Structure

- 6.0 Introduction
- 6.1 Objectives
- 6.2 Data Warehouse – Key Challenges
- 6.3 Data Lakes
 - 6.3.1 Need for a Data Lake
 - 6.3.2 Data Warehouse Vs Data Lake
 - 6.3.3 Data Lake Maturity
 - 6.3.4 Data Lake Architecture
- 6.4 Data Swamp
- 6.5 Complex Data
 - 6.5.1 Complex Data Modeling
 - 6.5.2 Complex Data Models
- 6.6 Cloud Data Warehousing
 - 6.6.1 Reasons for Migrating to the Cloud
 - 6.6.2 Challenges of Cloud Data Warehouses
 - 6.6.3 Building a Successful Cloud Data Warehouse
- 6.7 Real Time Data Warehousing
 - 6.7.1 Real-Time Data Warehouse architecture
 - 6.7.2 Real-time Data Warehouse Architecture Tradeoffs
- 6.8 Data Warehousing and Hadoop
 - 6.8.1 What is Hadoop?
 - 6.8.2 Hadoop Architecture
 - 6.8.3 Conceptual architecture of Hadoop Data Warehouse
 - 6.8.4 Advantages of Building a Hadoop Data Warehouse
 - 6.8.5 Challenges of Building a Hadoop Data Warehouse
- 6.9 Data Warehouse Automation
 - 6.9.1 DWA Maturity
 - 6.9.2 Data Warehouse Automation Tools
 - 6.9.3 Advantages of DW Automation:
- 6.10 Summary
- 6.11 Solutions / Answers
- 6.12 Further Readings

6.0 INTRODUCTION

In the earlier units, we have learned the conceptual aspects of a data warehouse including its types and their underlying architectures. You have also seen how new forms of databases evolved viz. Relational, Object-oriented, etc. which, in turn influenced the architectures of data warehouses to encompass the need for accommodating various forms of data. The earlier challenges of data warehousing were successfully addressed through research like innovative architectures, algorithms, modeling techniques, methodologies, etc..

Researchers have been trying to find better solutions for data warehousing and analytics for past few decades. The focus has been limited to relational technologies. Data management solutions began with databases (initially relational), then as the magnitude of data increased the world saw the advent of very-large-database systems (VLDBs). As data originated from various sources and consecutively possessed the directional attribute (via data-sharing and distributed storage, etc.) the research and technological focus evolved towards non-relational systems. Presently, we are in the phase of data–explosion called Big-Data and consequently there have been new innovations in this direction towards managing extremely huge magnitude of data without losing the essence of performance, scalability, reliability, security, etc..

In this unit we will discuss the trends in data warehousing namely data-lakes, complex data-marts, migration of data warehouses to cloud environment, real-time data warehouses, Hadoop software and how it supports data-warehouse and automated data warehouses.

6.1 OBJECTIVES

After going through this unit, you shall be able to:

- understand changing technological landscape related to data warehouse;
- describe data lakes, complex data marts and real time data warehousing;
- understand the concepts of Cloud-based data warehouse;
- describe how Hadoop supports the data warehouse design pattern;
- understand Data Warehouse automation and its necessity, and
- list examples that support the above data warehousing trends.

6.2 DATA WAREHOUSE – KEY CHALLENGES

When we attempt to understand a data warehouse from a technical perspective we comprehend that it is a database of considerable size ranging from several hundreds of gigabytes to several terabytes and in some cases, few petabytes too. The sheer enormity of database size, the complexity of analytical queries, data mining algorithms and the heterogeneous aspect of data loaded into a data warehouse makes the entire scenario incomprehensible and complicated. These results in significantly acute technological challenges for researchers while raising critical questions for which, solutions appear to be equally intricate or pragmatically irresolute.

It is evident that exhaustive research in data warehousing is going on in the areas namely, conceptual modeling of DWs and logical data models, data warehouse loading (data-refreshing), execution efficacy of OLAP queries and data mining algorithms, materialized views, data analysis techniques, metadata management, evolution management of DWs, stream-based, real-time and active data warehouses, and complex data warehousing (for example spatial, XML, object, multimedia).

Some of the research and technological challenges of utmost significance are dealt through research on topics like amalgamation of heterogeneous data sources, i.e., ETL, cleaning, source characterization, and data integration. Data source discovery, metadata management and standardization, handling data source evolution at an integration layer as well as data flow (ETL) performance optimization are topics

which still lack sufficient attention in research. Big data management has exposed many new challenges due to data-explosion, caused mainly by the diversity and velocity of data-generation. Solution development for the assimilation and storage of big data is imperative for design and implementation of a fully functional big data architecture that can be used by analytics, recommendation systems, visualization systems, etc.

In the earlier units we have discussed the challenges faced by researchers pertinent to data warehousing issues like complications in utilizing semi-structured and unstructured data from various sources, etc. It is evident that the exponential growth in data, including the numerous attributes inherited had a profound impact on the architectural elements of data warehouse. There are yet many capabilities and extensions of functionalities that necessitate their integration into a state-of-the-art system. The additional functionalities if developed severally, are required to function seamlessly as an integral part of the primary system. Architectural impact of issues requires critical thought for implementing solutions that offer optimizing total cost of ownership (TCO) with substantial return on investment (ROI). Since researchers have already embarked upon the journey to build newer architectures to address the proliferation of data and its capture to produce meaningful results, we are likely to see new trends emerging in the form of solutions and challenges for the future.

Let us look at the key challenges faced by researchers which have been categorized based on their technical significance.

Computational techniques: A primary candidate for this research challenge is the computational scalability required with drastic increase in data volumes. Emerging technological trends like Cloud data warehouses (CDWs) may address this issue. Parallel and In-memory computational techniques are being developed and perfected.

Design: Efficacy of design has been of particular significance for researchers. Due to performance implications involved design techniques need to consider the following questions carefully:

- a. How can design dependent latencies for cubes be minimized? For example, the computational costs of aggregated data in case of huge datasets may prove unreasonable.
- b. How can the update and refresh times for data warehouses (specifically cubes) be improved upon? For example, there exist various techniques and strategies for accomplishing these tasks which tend to fail for unstructured/heterogeneous data.

Quality: The word has different connotations in different contexts and is understood accordingly. For example if the query is complicated the underlying cubes would be overwhelmed with number of dimensions and measures required to build the cubes, especially if the data is unstructured/heterogeneous in nature. This can be attributed to the skill level of the data warehouse user. On the other hand, if even a simple query cannot fetch data in a fair amount of time then the modeling techniques may be questioned. If the modeling is impeded by lack of functionality then the architecture or design of data warehouse software itself is questionable. Therefore, quality plays a significant role at every stage of the data warehousing process. Good

governance and best practices may partially resolve the quality related issues but a holistic scrutiny would reveal any gaps if extant.

Size: The essence of practicality is eliminated when a data warehouse's computational capabilities are throttled. This is caused by the cumulative size of fact tables which tend to expand exponentially over huge datasets.

Operability: A data warehouse needs to provide seamless integration with external data sources for data capture and collection. Interoperability is also another challenge when communicating with various devices and software components. Data warehouses usually make use of Representational State Transfer Application Programming Interfaces (REST APIs) to overcome these issues but security risks increase with the use of such components.

In-memory representation: Distinct from the in-memory computation techniques mentioned above this issue deals with the efficacy of memory-based representation of the data warehouse and its components like OLAP cubes, datasets and result sets. For example, the increasing number of dimensions causes explosive increase in cardinal values for cells. Consequently, secondary and tertiary memory based solutions require critical analysis. SAP HANA which is a High-performance ANalytic Appliance multi-model database that stores data in its memory instead of keeping it on a disk. This results in data processing that is magnitudes faster than that of disk-based data systems, allowing for advanced, real-time analytics). SAP HANA is exemplar of such in-memory database implementation utilizing a cloud platform.

User Centric Interface: Like other tools mentioned above that are required for specific functions it is imperative that the interface allows the user to make optimal use of the data warehousing software. In order to extract maximum benefit (ROI) from the data warehouse implementation the user should not be exposed to the complexities of the system. This challenge has led to a trend of developing autonomous databases equipped with self-monitoring mechanisms and self-healing techniques. There is also a trend towards making as many data warehouse processes automated as possible with a goal of achieving zero maintenance. This is to reduce the maintenance burden on individuals and organizations. It also has other advantages like reduction in human error rates, saving time on maintenance tasks, etc. It also allows organizations to cut down on resources and maintain a lean environment. The user would only intervene if a business decision is involved. A user centric interface also reduces training expenditure while putting the controls at user's fingertips.

Pioneering Infrastructure foundation: Converged, Hyper-converged and Dynamic Infrastructures are deployed in data-centers. These are deployed either on-premises / onsite (private cloud) or on the Cloud (public cloud, e.g. Amazon Web Services (AWS), Google Cloud Platform GCP), Microsoft Azure (MSA), etc.) or a combination of both Private and Public cloud also called as Hybrid cloud platforms. These platforms can provide an innovative foundation for medium to exceptionally large data warehouse implementations. Also availability of specialized computing infrastructure components like graphic processing units, Artificial Intelligence (AI) or Machine Learning (ML) based Processor chips are available (example NVIDIA's Tesla or Drive PX series processors, AMD's Fire Stream or Radeon Instinct processors). The data warehouse software also needs to

improve its architecture and design constructs to derive the maximum benefit from such advanced infrastructures.

Complexity: For example building OLAP cubes over exceptionally large datasets attracts the penalty of rising complexity concerns that are usually unheard of in relational data warehouses. For example, there is an exponential increase in the number of dimensions in case of unstructured data in addition to several measures due to unstructured data.

Optimization and Innovations in Query Language: Traditional query languages like Multidimensional Expressions or MDX developed for OLAP databases was later wrapped in XML to be called mdXML to include XML data retrieval. With data undergoing more diverse forms and originating from equally varied sources the query languages require serious thought to improve their performance and capabilities to handle sufficiently large data from disparate sources while being able to build and populate cubes through query plan optimizations.

Data Visualization: Many organizations and institutions capture data from varied sources to assist in strategizing, business decisions, weather forecasts, predicting climate changes, knowledge visualization, cognitive maps, etc. In all these instances the sheer amount of data collected needs to be managed efficiently on various fronts to produce viable reports in order to visualize data. In healthcare systems that are dynamic in nature a patient's health parameters captured via various sensor devices need to be analyzed and evaluated in real-time. Other dynamic models like Climate change prediction systems need to monitor data captured via Internet of Things (IoT) devices are used to predict weather/climate anomalies or impending catastrophes. These scenarios require state-of-the-art software and infrastructure components. The algorithms implemented via software and computed using the infrastructure has to make sense of the data and at times discover hidden meanings and therefore new knowledge. This requires specialized query tools for example GraphQL, etc. while at the same time the semi / unstructured aspect of data requires appropriate management software that is architecturally robust. There are other critical issues with AI / ML based data visualization like necessity for innovative algorithms, multi-parameter based optimizations, accuracy of results, etc.

In the next section, we will learn about the Data Lakes which hold a large amount of data.

6.3 DATA LAKES

A data lake is a central location that holds a large amount of data in its native, raw format. Compared to a hierarchical data warehouse, which stores data in files or folders, a data lake uses a flat architecture and object storage to store the data. Object storage stores data with metadata tags and a unique identifier, which makes it easier to locate and retrieve data across regions, and improves performance. By leveraging inexpensive object storage and open formats, data lakes enable many applications to take advantage of the data

Data lakes were developed in response to the limitations of data warehouses. While data warehouses provide businesses with highly performing and scalable analytics, they are expensive, proprietary and can't handle the modern use cases most companies are looking to address. Data lakes are often used to consolidate all of an organization's data in a single, central location, where it can be saved

“as is”, without the need to impose a schema (i.e. a formal structure for how the data is organized) up front like a data warehouse does. Data in all stages of the refinement process can be stored in a data lake: raw data can be ingested and stored right alongside an organization’s structured, tabular data sources (like database tables), as well as intermediate data tables generated in the process of refining raw data. Unlike most databases and data warehouses, data lakes can process all data types — including unstructured and semi-structured data like images, video, audio and documents — which are critical for today’s machine learning and advanced analytics use cases.

6.3.1 Need for a Data Lake

First and foremost, data lakes are open format, so users avoid lock-in to a proprietary system like a data warehouse, which has become increasingly important in modern data architectures. Data lakes are also highly durable and low cost, because of their ability to scale and leverage object storage. Additionally, advanced analytics and machine learning on unstructured data are some of the most strategic priorities for enterprises today. The unique ability to ingest raw data in a variety of formats (structured, unstructured, semi-structured) along with the other benefits mentioned, make a data lake the clear choice for data storage.

6.3.2 Data Warehouse Vs Data Lake

A data warehouse stores structured business data in its processed form. This approach requires fairly rigid schemas for well-understood types of data. While data warehouses are an important tool for enterprises to manage their important business data as a source for business intelligence, they don’t work well with unstructured data.

Data lakes allow the storage of raw data; both relational, as well as non-relational that is intended to be used by data scientists and developers along with the business analysts. They take the data out of the silos and make it accessible to all business users promoting centralization of data. The key differences are summarized in the Table 1 given below:

Table 1: Data Warehouse Vs Data Lake

Attribute	Data Warehouse	Data Lake
Type of Data	Structured data from sources like transactional systems and operational databases.	Raw Data from varied sources like websites, mobile apps, IoT devices, social media channels etc.
Schema	Schema-on-write	Schema-on-read
Intended users	Primarily business analysts	Data scientists, developers and business analysts
Price/Performance	Fastest query results using higher cost storage	Query results getting faster using low-cost storage
Data Quality	Highly curated data that serves as the central version of the truth	Any data that may or may not be curated (ie. raw data)
Type of analytics	Business intelligence, visualization and batch reporting	Machine learning, predictive analytics, profiling and data discovery.

Attribute	Data Warehouse	Data Lake
Agility	Fixed configuration, less agile	Highly agile, can be configured and reconfigured as per requirements.
Security	More secure storage	Higher accessibility makes ensuring security a challenge

6.3.3 Data Lake Maturity

A data lake is said to go through various stages of maturity during its life-cycle, considering that it is a fairly new design pattern. Bill Inmon (the father of data warehouse) proposes data lake maturity process (or stages) as below:

According to **Bill Inmon**, native or raw data is classified into three discreet categories viz. i) Analog data, ii) Application data and iii) Textual data. He states as below:

“In order to organize the different types of data into a structure that can be analyzed, it is necessary to create a high-level structure of data within the data lake. As data enters the lake it first enters the raw data pond. The purpose of the raw data pond is to serve as a holding cell. There is little or no analysis or other organized activity of the data while in the raw data pond. Once it is time for analysis, the information in the raw data pond is sent to one of three different ponds based on the kind of data entailed. For example, analog, application and textual data all require a unique data pond.”

When the data has passed its useful life in the data pond it is moved from its respective data pond to an archival data pond as illustrated in Figure 1.

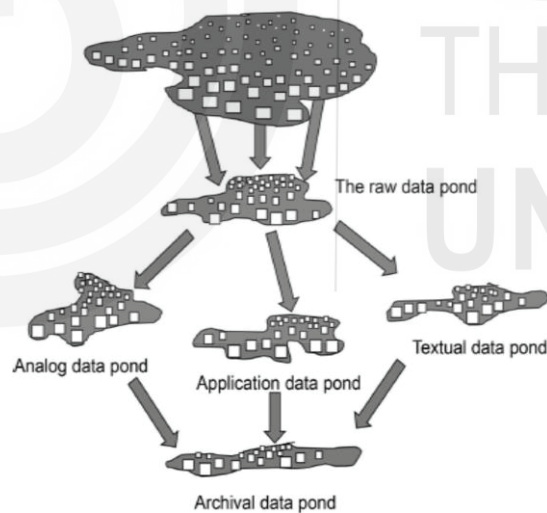


Figure 1: Data Lake Architecture by Bill Inmon

Alternatively **Alex Gorelik** introduces additional data lake stages, as part of data lake’s maturity aspect, as below:

- a. **Data Puddle:** A data mart built using big data technology with a sole purpose or as a single-project with data being loaded into it for the consumption of a single project or team. The data mart concept is well known and use of big data technology is mandated to reduce cost and improve performance.
- b. **Data Pond:** Similar to that proposed by Bill Inmon, except in this case it is a collection of data puddles in the form of collocated data marts or similar

to a poorly designed data warehouse with minimal transformation of source data. As such these data ponds restrict data usage solely to the projects that necessitate it.

- c. **Data Lake:** It caters to business users in two significant aspects. Firstly as a self-support service through which business users are able to utilize the data and secondly, provisioning data to business even when not required by the project.
- d. **Data Ocean:** Facilitate enterprise-wide availability of self-service data and data driven decision making, irrespective of the data location or its existence within the data lake.

The above components are shown in Figure 2 depicting data lake maturity process proposed by Alex Gorelik.

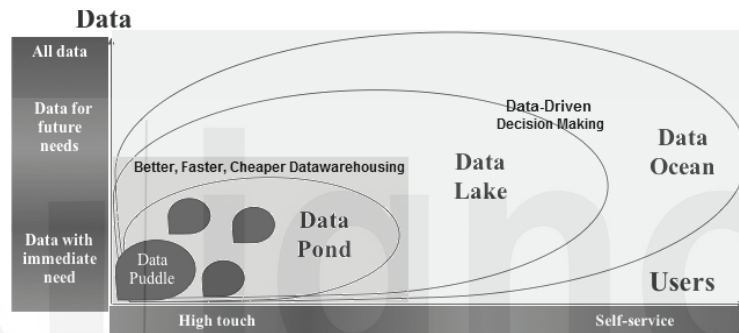


Figure 2: The Enterprise Big Data Lake by Alex Gorelik

6.3.4 Data Lake Architecture

The below given Figure 3 shows a standard proposal for an architecture of a data lake system. The system consists of four layers which are i) Ingestion Layer, ii) Storage Layer, iii) Transformation Layer and iv) Interaction Layer.

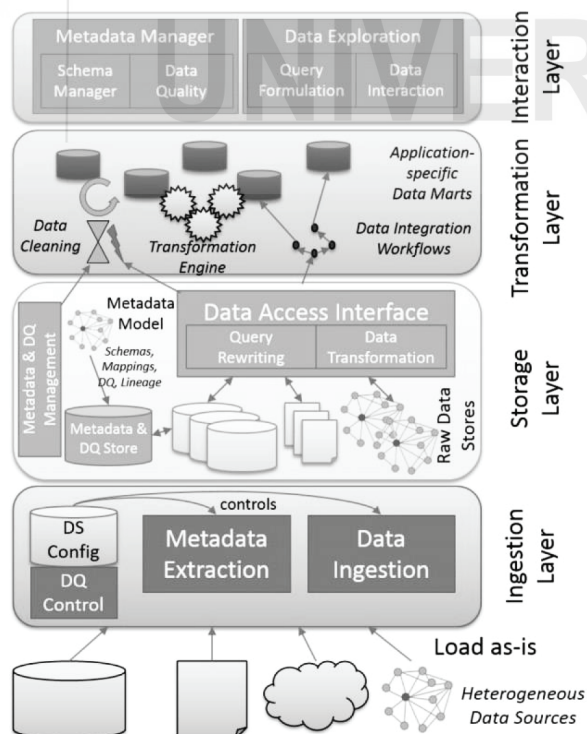


Figure 3: Layered Architecture of a Data Lake

i) Ingestion Layer

The Ingestion Layer is in charge of bringing data into the data lake system from various sources. One of the main features of the data lake concept is the ease with which any type of data can be ingested and loaded. Data lakes, on the other hand, have been repeatedly reported as requiring governance in order to avoid becoming data swamps. Administrators of data lakes are in charge of this critical topic.

The metadata extractor is the most key component of the ingestion layer. It should make it easier for data lake administrators to configure new data sources and make them accessible in the data lake. To accomplish this, the metadata extractor should extract as much metadata as possible from the data source (for example, schemas from relational or XML sources) and store it in the data lake's metadata storage. The raw data, in addition to the meta- data, must be absorbed into the data lake. Since the raw data are kept in their original format, this is more like a "copy" operation, which is certainly less complex than an ETL (Extract-Transform-Load) process in data warehouses.

ii) Storage Layer

The metadata repository and the raw data repositories are the two key components of the storage layer. Since raw data repositories must be stored in their native format, data lake environments must support a variety of storage systems for relational, graph, XML, and JSON data. Hadoop appears to be a strong candidate for the storage layer's basic platform. To support data fidelity, however, additional components such as Tez or Falcon are needed.

iii) Transformation Layer

Data can be transformed from storage to user experiences using the transformation layer. It requires operations such as cleansing, format transformations, and so on.

iv) Interaction Layer

All of these functionalities that are needed to work with the data should be covered by the interaction layer. These functionalities should include visualization, annotation, selection and filtering of data, as well as basic analytical methods. It's also worth noting that more advanced analytics like machine learning and data mining should not be regarded as part of a data lake system's capabilities.

6.4 DATA SWAMP

A data swamp is a data pond that has expanded to the size of a data lake in the absence of self-service and governance facilities. At best, the data swamp is used like a data pond and at worst it is left unused. While various teams may frequently use areas of the lake for their projects the bulk of the data is unclear/more difficult to understand, undocumented and therefore, unusable.

Data swamps can be accessed by those who are technologically proficient. This is accomplished by incising or chiseling small puddles for themselves and their teams. At the advent of data lakes many enterprises hastily bought Hadoop implementations and loaded them with petabytes of raw data, from various sources, without clarity on the manner of its utilization. In the absence of any systematic approaches to organize the data while loading it turned into an incomprehensible mess. Additionally, enterprises were prohibited by governance regulations against

permitting data swamps’ access to a wider audience without obfuscating sensitive data. Since the precise location of sensitive data was dubious, access was interdicted. Consequently, the data essentially remained unusable and unutilized.

☞ **Check Your Progress 1**

- 1) What are present challenges in data-management?

.....

- 2) Describe in your own words the logical data lake concept?

.....

- 3) What is your interpretation of a successful data lake?

.....

6.5 COMPLEX DATA

We have already seen how big data has become pervasive across enterprises. We have also seen different ways in which data is generated through various sources like emails, files, IoT, Logs, etc. Enterprises. With the scale and speed of data adding to the type of data being generated, the term Complex data was coined to define all data that does not conform to standard data types like dates, currency, alphabets, numbers, etc. Complex data is still essential, as we have seen when dealing with raw data, in building data-insights and big data analytics. Complex data is processed as complex datasets and is used by pattern matching algorithms in Machine Learning models. Here we take a quick look of how the complex data trend has influenced the data-driven technologies and enterprises.

6.5.1 Complex data modeling

With data streams traversing many hubs and myriad technologies complex data modeling have become a standard practice across the industry. Complex data travels through data processing tools, ETLs, ERP software, data lakes and other paths. Complex data has its intrinsic traits like its own syntax, schema, technology, terminology and type. This diversity of traits complicates the work of data modelers.

Various models like statistical, polyglot, no payload, multi-level, etc. exist to store, process and analyze complex data. Some standard models are also available like Anchor models and Data Vault Models. Both of these models provide advantages like Scalability, Temporal data handling and are resilient to structural and content-based changes.

6.5.2 Complex data models

6.5.2.1 Anchor Model

An example of the Anchor Model is shown in Figure 4. Anchor modeling has four elements – Anchors, Attributes, Ties and Knots.

Anchors

The anchors are used to model entities and events.

Attributes

Attributes are used to model properties of anchors.

Ties

Ties are used to model the relationships between anchors.

Knots

Knots are used in the modeling of shared properties, such as states.

Attributes and ties can be versioned (termed as historization), when changes in the information the model need to be retained. The different graphical symbols, representing the elements of the model, are similar to those used in entity-relationship modeling with a few extensions. An outline on a tie or attribute depicts versioning of changes.

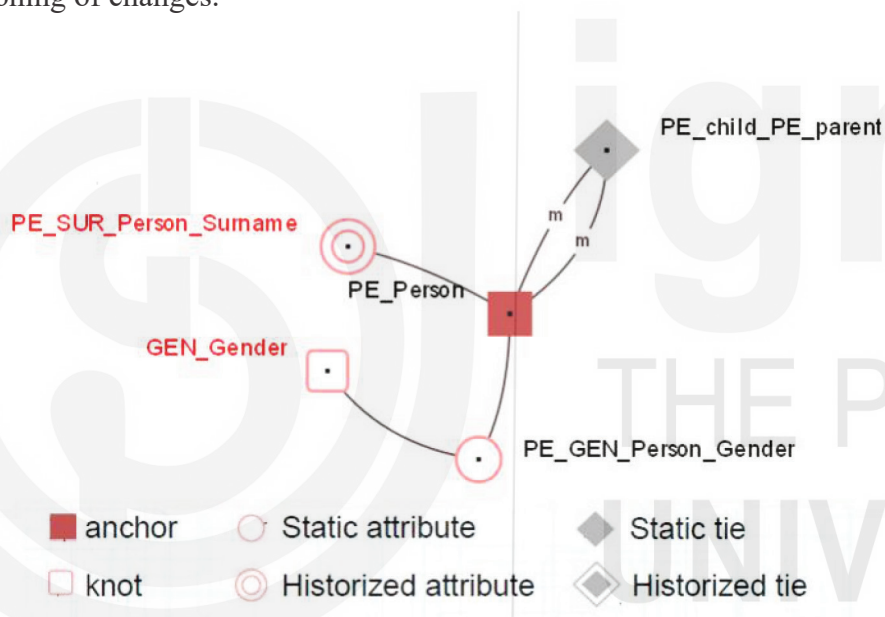


Figure 4: An Example of the Anchor Model

6.5.2 Data Vault Model

Data Vault is a database modeling technique offering long-term storage of historical data arriving from multiple operational systems. The model also allows coping with issues such as auditing, data traceability, data loading performance and resilience to change.

Data traceability implies that every row of data in a data vault must be associated with a record source and date-of-load attributes. This information allows an auditor to trace values to their respective sources. Daniel (Dan) Linstedt developed the Data Vault model in 2000. Figure 5 illustrates the Data vault model. The components of this model are briefly discussed below:

Hubs

Hubs are comprised of a list of distinct business keys with low affinity to change and contain a surrogate key per Hub item. The Hub also contains metadata designating

the source of the business key. Satellite tables; discussed below, store descriptive attributes for the data on the Hub (for example, a multiple languages description for the key). A Hub should have at least one satellite and at the least, should comprise of the following attributes:

Surrogate key: Connects the other structures to the table comprising the surrogate key

Business key: The driver attributes of the Hub structure may contain multiple fields.

Record source, Allows determining which system was the first to load a business key

Metadata (optional) Contains information on manual updates (user/time) and data extraction date.

Existence of multiple business keys is precluded in a Hub, except when two systems provide the same business key but having different meanings to resolve conflict.

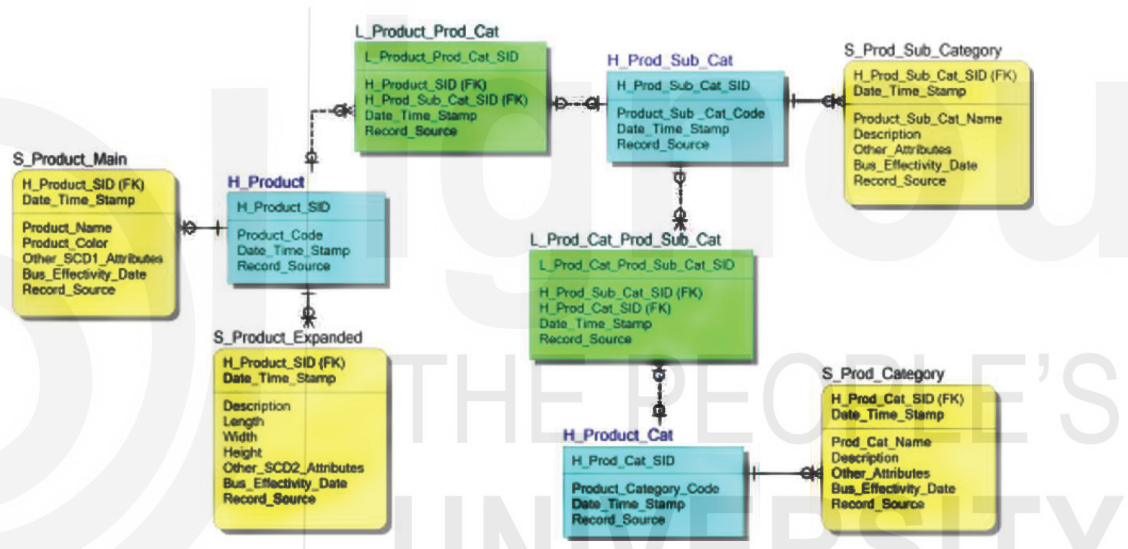


Figure 5: An example of a simple Data Vault model with hubs (blue), four satellites and links (green)

Links

Link tables are models providing relations between business keys and are fundamentally many-to-many join tables, containing some metadata. Links can connect to other links, to handle granularity variations. Using link references within another link is bad modeling practice, as it creates dependencies between links making parallel loading of links more challenging. Sometimes links connect hubs to data that is not adequate for hub creation. This scenario happens when a non-real business key is associated with the link. This happens only when using the business key for another link or as key for attributes in a satellite is not possible. This technique has been called a 'peg-legged link' by Dan Linstedt on his (now defunct) forum. Links also comprise of surrogate keys for linked hubs. The surrogate keys of the hubs, for the link and metadata describe the relationship source. As in hubs, the descriptive attributes for the relationship information gets stored in the satellite table.

Satellites

The hubs and link's temporal attributes and descriptive attributes stored in separate constructs called satellite tables or simply satellites. The satellites also comprise metadata connecting them to their parent link or hub, metadata designating the relationship source and attributes, also including a time-series with the attribute's start and end dates. While the links and hubs deliver the skeletal structure of the model, the satellites deliver the "meat or substance" of the model in the form of business processes contexts that are stored in links and hubs. Storage of these attributes is done relative to the timeline and content details and span from relatively simple (a linked satellite with only a timeline and a valid indicator) to relatively complex (all fields designating a client's entire profile).

Typically, the source system groups attributes in satellites. Rate of change for descriptive attributes like amount, size, color, speed or cost may vary. These attributes can be separated into various satellites based on the rate at which they change.

Metadata is contained within all the tables, designating the validation-date of the entry and system from which it originated, providing a holistic historical perception of the data as it is loaded into a data warehouse.

Check Your Progress 2

- 1) Explain your understanding of complex data and the need for complex data modeling.

.....

6.6 CLOUD DATA WAREHOUSING

The transformation of raw data into insights is a challenge and the solution is not simple. That challenge is made worse by the accelerated pace of environmental dynamics affecting the enterprises. The pace at which business is done is in fact accelerating. Things that we expected to be done in a week or overnight or an hour need to be done in real time. Innovations in various industries are taking place simultaneously or at an overwhelming pace.

The challenge for most enterprises is to discover opportunities to exploit this change rather than being overpowered by it. The answer, though superficially simple, lies in leveraging data and transform into a data-driven enterprise. This transformation should not be limited to a technological perspective, rather it needs to be assimilated and trusted in by people across the enterprise as the crux of organizational health and wellbeing.

Enterprises have to confront demands of space and location (capacity to store and utilize voluminous data. Further, new demands imposed by regulatory and compliance mandates, data velocity, new data sources, performance, scalability, QoS, etc. have compelled the enterprises to rethink their data-leveraging strategies.

Enterprises have realized that a data warehouse implementation is no more a silver bullet for their problems considering the environmental dynamics and other issues discussed above. Therefore, we have recently witnessed the trend of extant data warehouses migrating to cloud while new ones are adopting cloud as their data

warehouse implementation platform. This forethought is a consequence of realizing issues of capacity planning, infrastructure costs, usage policies, governance, flexibility of utilization based expenditure and many more.

6.6.1 Reasons for Migrating to the Cloud

On-premises data warehouse implementations are reducing and consequently its business is diminishing too. Vendors like Google BigQuery, AWS Redshift, Snowflake, Azure Synapse Analytics, etc. are popular ones for cloud hosted data warehouses. These vendors offer near-instantaneous installations with storage, computing and network resources with increased performance and scalability benefits. The supporting software and platform are always maintained up to date. Most enterprises or a new data lake tend to choose a cloud-hosted data repository.

Enterprises are migrating to cloud data warehouses because they get instant access to all the infrastructure resources necessary to scale that solution. They also derive a pay for use benefit as they pay only for additional memory, compute, storage, networking or other resources they might need to scale. Best-of-breed applications can also be availed through SaaS to their cloud-implementations, whereby their entire business intelligence and analytics stack can be provided as a service. They can scale up or scale down as required with minimal installation or maintenance charges. The trend towards cloud-based data warehousing is clearly evident and that is where this market is progressing.

To sum up, enterprises are adopting the cloud migration path due to the following advantages offered by a cloud platform:

- i. Improved performance
- ii. Flexibility of cost through flexibility of cloud resources utilization
- iii. Migration of existing products to the cloud environment
- iv. Derive benefits that are inherent to a cloud platform

6.6.2 Challenges of Cloud Data Warehouses

Although cloud-based data warehouses (CDWs) are an essential component of the future of enterprise-data, there are additional tasks to be performed and therefore, challenges involved. The significant challenges of hosting data warehouses in a cloud environment are as below:

- i. Data extraction, transformation and subsequent loading
- ii. Context and User based Data access
- iii. Management of heterogeneous data velocity
- iv. Ensuring instant availability of new data sources
- v. Ensuring data quality
- vi. Management of sensitive data and compliance to regulatory mandates
- vii. Interoperability with tools and infrastructure external to cloud environment
- viii. Communicating with legacy systems that could not be migrated to cloud for technical and organizational reasons
- ix. Data governance and obfuscation of sensitive data
- x. Automation of data offload for sophisticated analytics and ML

6.6.3 Building a Successful Cloud Data Warehouse

Building a successful Cloud Data Warehouse (CDW) entails overcoming the challenges mentioned in the previous section. Since cloud data warehouses deal with enterprise data at scale, eliminating or reducing the impediments in its path to derive maximum business value forms a crucial element in building a successful CDW. To ensuring a successful CDW implementation the CDW has to go through the following stages.

Stage 1 - Formulating an approach for data curation at scale and data integration

Firstly, enterprise resource scalability is imperative with increasing volumes of data that results from business growth. A precise and pertinent data curation and integration approach is a deciding factor in an enterprise's ability to succeed as a data-driven enterprise. This success in turn is based on an enterprise's ability to leverage data analytics based insights ahead of its competition.

In order to carefully plan the approach to data integration the following considerations are essential:

- i. Preparedness for enterprise changes, besides technological changes, or the ability to prepare for such scenarios.
- ii. Enterprise readiness to adopt new technologies including unknown ones developed in the future
- iii. Future objectives of your enterprise that would have a critical impact on your data integration architecture already in place
- iv. Enterprise expectations from this approach in terms of adding measurable and tangible value to its business growth

The above considerations entail due diligence in solution implementations as below:

Transformation into a data-driven enterprise involves changes on the technological front but also at various other levels. These changes affect both the technical and business stakeholders alike. To list a significant few, the changes involve enterprise risks, costs, dependencies, ROI, benefit-amendments, assumptions, and cultural issues. For large enterprises, this necessitates the presence of change management professionals, who educate, train and influence stakeholders mindsets to the imminent changes. Enterprises planning to undergo a data-driven transformation may hire change-management agencies or professionals based on the scope and dynamics of their businesses growth.

Architecture of systems employing data integration technologies should be flexible enough to accommodate new tools in the future with minimal impact on the architecture and consequently the system services. Conceptualization of future-proof architectures is essential to make a system capable to adapt to yet unknown technologies that might be developed in the future.

Enterprises may have plans to expand their data-leveraging capabilities, for example moving their Enterprise Resource Planning (ERP) data i.e. data from ERP modules like Customer Relationship Management (CRM) / Supply Chain Management (SCM) / Human Resource Management (HRM), etc. to the cloud data warehouse. However, the actual objectives of the enterprise might involve efficient master data or workflow management to emerge as a leader in customer services landscape.

Such objectives need to be stated upfront in order to formulate the best approach that caters effectively to present and future business requirements.

Wisdom dictates that expenditure on the latest and greatest technological tools might not be adequate for business growth if they cannot cater to the specific functions of an enterprise. Also, availability of great feature-sets in a tool does not necessarily guarantee business growth if only a few of them can be truly exploited to an enterprise's advantage. Tools that offer customization or flexibility in terms of specific functionality needs, choice of features, etc., in a data integration platform are available. A clear understanding of the customizations involved allows technology decision makers to understand the implementation components required to collate data and ensure its availability across the enterprise. It also helps the enterprise understand TCO of the data integration platform, which helps to optimize and improvise the approach to extract maximum ROI. Further, the approach also enables a precise understanding of the business benefits expected from a data integration platform and subsequently, their measurability and tangibility.

Stage 2 – Leveraging Data Integration platform for on-demand data provisioning

It is imperative for an enterprise to develop faster decision making capabilities. However, the vast volumes and heterogeneous characteristics of data make it extremely difficult to do so if it is not able to scale its infrastructure and tool-sets to match the rate of data explosion (i.e. rate of increase of data velocity, diversity and volume). Data-explosion also triggers the need to assimilate and utilize data at equivalently faster rates i.e. before its business worth diminishes. The fresher the data is, the most up-to-date the data visualizations, which give an almost instantaneous view of the most recent business landscape. These representations allow an enterprise to make informed decisions to strategize, improvise, adapt and steer their business on the path to success.

To extract maximum value from data, the right data should be available to right people, and at the right time. After the enterprise has sourced huge volumes of data it may develop the capability to store. This data needs to be transformed based on business requirements so that it is available to various people with specific expertise related to business areas. For example, data engineers should have access to raw data, analysts should have access to curated and organized data (e.g. in relational format), and other business functions like human resources, finance, manufacturing etc. should have access to their respective data sets. Sensitive data would also require access controls and obfuscation for cross functional requirements. All this advanced work will necessitate time and resources, creating latencies at different levels. The latencies at different levels like data extraction, data storage, organization and management, access control, access to desensitized data, transformation, loading, etc. Also, if the data cannot be visualized at scale through the use of appropriate tools the business decisions get delayed. It is easy for us to understand now; that in order to reduce the machine level latencies a meticulously planned data integration approach is imperative.

An approach to a fully optimized data integration platform to supplement business data needs of business will allow data-sharing across the enterprise, irrespective of functional domains and thereby, enhancing data visualizations through combined insights. Data provisioning puts data at the fingertips of stakeholders with very little dependence on technical personnel thus allowing for self-service. As a result of self-service capabilities operational latencies are drastically reduced while at

the same time increasing innovative data insights. As ideas and representations are shared across functional domains, they can be improved upon through collective inputs and critical thinking. A thoughtfully planned approach should consider the advantages within the boundaries of data governance.

Stage 3 – Ensuring high quality of heterogeneous data across CDWs

As already mentioned, in Stage 1 above, utilizing a vast collection of the best tools does not guarantee best results if the tools cannot be customized to the business requirements of an enterprise. The technical dependencies cannot be resolved in case of some tools that offer deep customizations and a host of features (of which, all might not be applicable). Conversely, tools that are easy to implement may have limited capabilities and therefore cannot be classified as enterprise applications and their implementation, despite, is likely to cause more operational latencies than if not implemented at all. Therefore, it is necessary to scrupulously evaluate and scrutinize various tools, for your data platform so that they can meet the current and future needs of the business. Ensuring high quality of data availability, ubiquitously across your CDWs, is now possible through cloud-aware ETL tools which people with basic technical knowledge can easily use. The cloud based ETL tools are compliant to security and regulatory requirements and employ state-of-the art Artificial Intelligence or Machine Learning techniques making the data reliable and accessible. As discussed in Stage 2, data sharing is necessary for timely decision making through mitigation of operational latencies. Implementing the right set of tools across different functional domains can expedite analytics and decision support capabilities.

☞ **Check Your Progress 3**

- 1) What are the reasons for migrating to a Cloud Data Warehouse (CDW)?

6.7 REAL TIME DATA WAREHOUSING

What follows on from the discussion on various stages of implementing a successful cloud data warehouse, in the above section, is the most essential and significant attribute of a data-driven enterprise, which is timely availability of data across the enterprise. Timely or instantaneous data availability combined with appropriate access has many connotations, based on performance, context, relevance, perspective, etc. The rationale behind this expectation is that in a data-driven enterprise, data undergoes various stages before it can be available to the management in the form of meaningful representations.

The representations or data visualization reports need to be based on trustworthy and most current data to be of any worth to the business. To make this possible the underlying processes – of sourcing the data, transforming it and making it available to various functional domains across the enterprise – need to happen as optimally as possible within the shortest possible time.

A birds-eye view or a look at the bigger picture makes it evident that the data platform that drives the enterprise should essentially work on-time across every stage of the journey of data. If we look at the detailed level, we realize that every element expects data to move immediately or in real-time.

The laws of science do not make the real-time concept possible; therefore, real-time terminology is sometimes altered to near-real-time. Enterprises set performance thresholds for their real-time interpretation, and if data warehousing capability falls within these thresholds, it is termed a real-time data warehouse (RTDW). The proliferation of new data or updates to actual data across a data warehouse immediately has become synonymous with real-time data warehousing and is the favored definition in the industry.

CDWs are an excellent candidate for enterprises wanting to implement real-time data warehousing because of their intrinsic feature to scale resources e.g. computational, memory-related, storage, networking, etc. Despite the best resources available, processes like data transformation, and data loading into a dimensional model are distinctively complex. Also, taking into account the massive amounts of data involved, the complexity gets further compounded. Therefore, the processes for updating new data coming in from different sources require specialized handling to make the latest data available in real-time.

We already know that the primary component of data warehousing i.e. the ETL process takes care of data movement from its source to the data warehouse storage and is extremely resource intensive. The mandates of real-time data warehousing require drastic alterations to the way ETL process takes place in addition to other components of the data warehouse.

Researchers have come up with some unconventional approaches e.g. novel ETL architectures, update methods – cached, partial, separate synchronized DWHs, etc., to ensure data availability in real-time. Figure 6 is a conceptual representation of real-time big data analytics which gives us a glimpse of what it means to implement a real-time data warehouse.

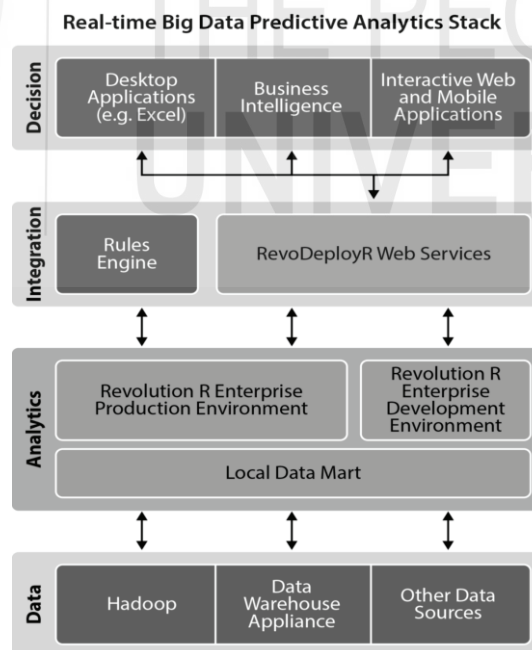


Figure 6: Real-Time Big Data Analytics: From Deployment to Production

Hadoop is a popular choice at the data-layer as it offers many tools for ingesting and transforming data in real-time. It also allows for integration of other tools like Spark for replacing the built-in MapReduce tool. For example, the Spark engine that powers the Shark (a RTDW) data warehouse gives a 10-100 times faster

performance than MapReduce. The Analytics and Decision layers are bound together by the Integration Layer and this composite collection of layers is fed through various components of the data layer.

6.7.1 Real-Time Data Warehouse Architecture

The Integration Layer binds together the Analytics and Decision layers and this composite collection of layers is fed through various components of the data layer. We can see that these changes expose a typical constructs that are essential for the way

- a) data sources are added/updated,
- b) data is propagated across the warehouse and
- c) data is loaded into various repositories for heterogeneous data.

We will discuss a few architectures that were proposed over the past decade and have greatly influenced some of the popular data warehousing components today.

In the Figure 7, a Near Real Time Data Warehouse Architecture is shown which comprises of following component blocks:

- i) Data production systems hosted on data sources and responsible for populating the data warehouse. This block comprises of a source flow regulator (SFlowR) that periodically (based on preset or customized policy) transmits data to the data warehouse after the data has been identified for pertinent changes.
- ii) The data processing area (DPA) responsible for data-quality and transformation. This block contains the data flow regulator (DFlowR) used for detecting the transmission-ready source.
- iii) After cleaning, an ETL workflow collects the records from the transmitting data source and transforms them to data warehouse format. The DPA also has other myriad responsibilities like ensuring QoS, checkpoint generation, caching data into reservoirs if the data warehouse pipelines are full, etc. Post-processing, the data transmission to the data warehouse, is orchestrated by a workflow regulator (WFlowR) component.

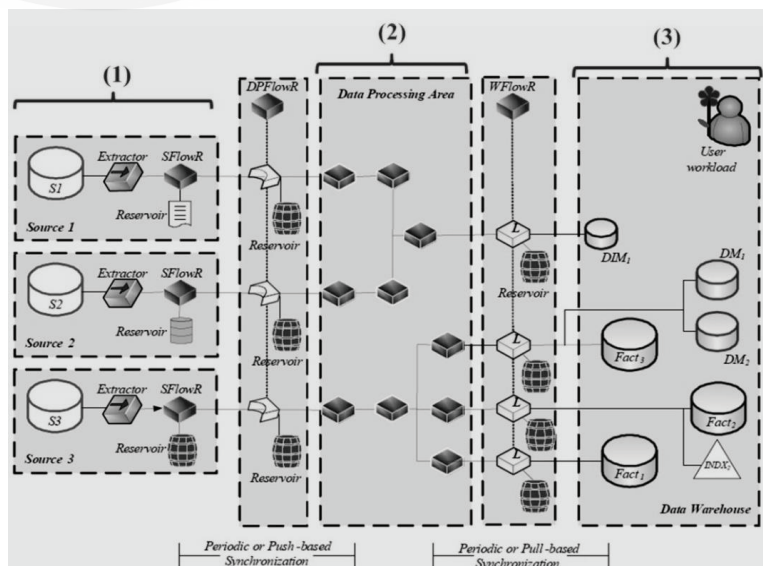


Figure 7: Near-Real-Time Data Warehouse Architecture

- iv) The data warehouse itself. The data warehouse is comprised of dimensions, fact tables, materialized views and indexes (primarily bitmaps and B+ trees) which makes it a complex repository. Data propagation through this block depends on the flow regulation components and the real-time aspect is ensured through the successful performance of these components.

An alternative architectural construct is proposed by Obali and his team is shown in Figure 8. The architectural components are: a) Metadata b) Web Service (WS) client, c) Web Service provider, d) ETL e) Real-time Partition, f) Real-time data integration and g) Data warehouse.

Change Data Capture (CDC) designating the modified data is initiated by the WS client and sent to the WS provider through a related web service. CDC involves use of various techniques for identifying changes to data like: direct extraction of source data if it is new, timestamp-based, application aided, file comparison based and trigger criteria based data capture. Data is populated in the real-time partition through a WS provider web service based on the data transmitted by WS client. The received data is separated into data and metadata. A structure query language (SQL) is generated, by a web service through SQL generator, using metadata and inserting data into data warehouse log tables. The generated SQL is executed to populate the data warehouse data repository (database).

The data orchestration between the data warehouse and real-time partition is handled by the real-time data integration component which acts as glue between the two. When there is a user-request for historical data the real-time data integration component generates a query that is sent to the data warehouse. In case of a hybrid data request (both historical and recent) a query is generated which integrates the data from the real-time partition and the data warehouse.

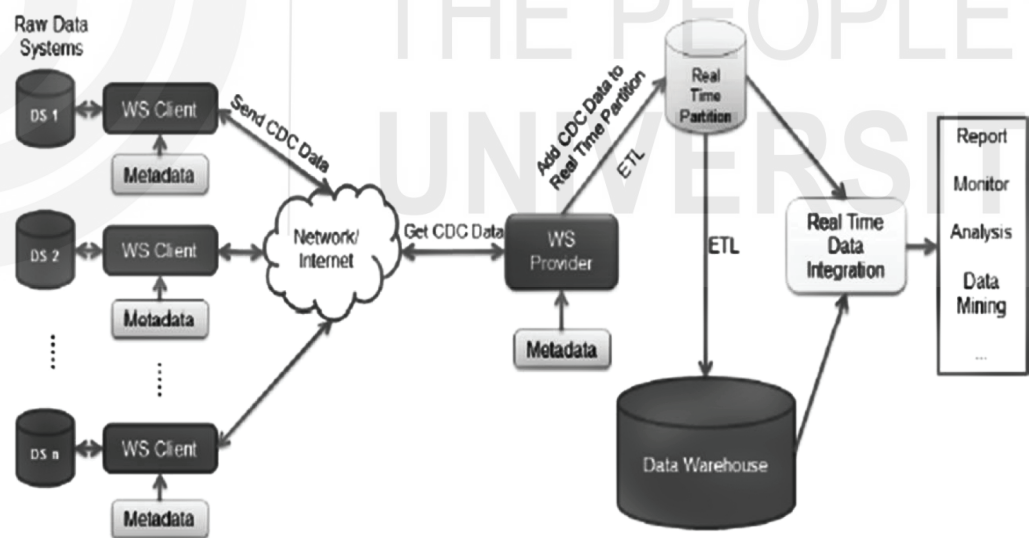


Figure 8: Real-Time Data Warehouse Architecture based on Web Services

6.7.2 Real-time Data Warehouse Architecture Tradeoffs

It is understood from the above architectures, few changes required across the entire data warehouse ecosystem to implement a real time data warehouse. Consequently, there are a few architectural tradeoffs that users of RTDWs need to understand.

The principles of ETL system are presumed to remain unchanged on the journey to a RTDW. Also responsibilities towards data integration, backup, recovery, archival,

regulatory compliance, security, quality, governance, etc. are to be maintained. The trade-offs that ensure on the RTDW journey is briefly explained as below:

Batch file replacement

Use of message queues and transaction log files are recommended by Ralph Kimball, replacing batch files. From a batch file's perspective, the data is complete with all keys resolved. Data from message queues and transaction logs is raw data that does not fall within the purview of business rules or corrective processes and is instantaneous.

Restrict scrutiny of data quality

Here the scrutiny is applied to columnar data. As data requests start to become more demanding, some data quality will need to be compromised by decreasing mandates on data quality. While this is done, users will need to be made aware of data unreliability and may require compensatory measures in the form of corresponding batch oriented ETL pipeline that periodically replaces real-time data with complete and accurate data.

Publish Facts with Dimensions

Old records of dimensions should be allowed to contain facts that are received early. For example, transaction details for a customer might be available even before a customer or item code is generated. We notice that the facts are available prior to dimensional data and in such scenarios if data for dimensions cannot be determined (due to real-time imperatives) an older version of the dimension must be used or a generic template could be used, otherwise. The dimension data should be posted into the hot (real time) partition as and when revised data is received or through the batch update scheduled for the end of day. Nevertheless, users should be informed that dimensions may contain transient data disparate from the fact data.

Preclude Data-Staging

Enterprise information integration (EII) systems stream data directly to the user's interface (screens) from the production sources, thereby precluding its transfer to ETL pipeline's permanent repository. Such circumstances, if extant in your enterprise, should be discussed with the senior management to appropriately assign responsibility for backup, recovery, archival, security, etc. of the uncommitted data.

Real-time partitions for Data Visualization

The demands for real-time data have increased with the decreasing time intervals of data transition between DW/Business Intelligence (BI) teams and production transaction processing. A real-time partition design as an extension of traditional data warehouse is a possible solution for resolving the increasing demands for data in real time. The real-time partition needs to conform to the following requisites:

- Contain complete set of activities since the last update of the static data warehouse
- Connect precisely as a true physical fact-table partition to the static DWs grain and content
- Allow continuous trickling of data through simple indexes. The ideal possibility of real-time partition being completely indexed may not be

viable in case of relational databases where indexes are logically disparate to the partitioning structure.

- Mapping real-time partition in memory to support high performance queries even without extant indexes.

Real-time partition for Transactions

Considering that the granularity of a fact table within the static DW is at transactional level it will contain a single row per transaction. The dimensional structure of real-time partition would be similar to the SDW and may not have any indexes as it requires continuous data loading. The real-time partition would also lack time series as it is supposed to contain only day's data.

For a comparatively large enterprise in the retail business with approximately 10 million daily transactions, the fact table would be sizable in volume. Assuming that the width of each transaction grain is 44 bytes (eight dimensions, three facts and four byte columns) you would have 440 MB of data daily. That would mean, yearly, about 160 GB of raw data in the fact table with heavy indexing and aggregations. However, the daily slice of 440 MB of real-time data should be pinned in memory. The real-time partition may have affinity for fast-loading and demonstrate high speed query performance simultaneously.

Real-time partition based on periodic snapshots

If we consider the same example as above but with a monthly granularity then the real-time partition would constitute the current active rolling month. Considering a customer-base of 15 million customers, a 24 month time series would accumulate 360 million rows in its fact table that would again be massively indexed with supporting aggregations. The real-time partition is a snapshot of the currently progressing month that is continuously updated as the month advances. Amendment of fully summative facts and semi-summative balances would be done on availability of respective data. The fact table superset of customer types across the enterprise would be quite narrow with say four facts and five dimensions giving us a real-time partition of 600MB allowing the pinning of the real-time partition in memory. With the rest of the monthly data arriving on the last day of the month, the real-time partition could be merged as the latest month onto the comparatively stable fact table. Then the process could be iterated for the subsequent month by flushing the real-time partition.

☞ **Check Your Progress 4**

- 1) How is a real-time data warehouse different from a traditional data warehouse?

.....

.....

- 2) Why is CDW a good candidate for real-time data warehousing?

.....

.....

6.8 DATA WAREHOUSING AND HADOOP

The title of this section may sound confusing at first but as we explore the Hadoop system in detail, we should be able to gain more clarity as to why data warehousing and Hadoop necessitates a combined discussion. Although the data warehousing and Hadoop trend evolved sometime back and has become a mature topic today it is still a continuing trend and therefore, requires an overview to comprehend the concepts involved.

Before we dive into the specifics of the Hadoop system we need to go through a few preliminary concepts and understand them so that we can discuss this topic of data warehousing and Hadoop without any confusion or doubts. We have already studied Data warehouse in the previous chapters above. In this section we will understand the definition and features of Hadoop. We will also compare and contrast Hadoop with a Data warehouse system. We will see how Hadoop can supplement and support a DW. Finally, we will take a look at the advantages and challenges involved in Hadoop supported DW.

6.8.1 What is Hadoop?

Hadoop is an open-source java-oriented framework comprising two fundamental components: i) Storage and ii) Processing and was developed, especially, to deal with massive volumes of data in a distributed computing ecosystem. The storage component of Hadoop is called the Hadoop Distributed File System (HDFS) while the processing component is termed as MapReduce.

Essentially, Hadoop, today, comprises of the following modules:

1. The Hadoop Distributed File System

The HDFS allows data storage with easy accessibility. The data storage resides in an environment guaranteeing highly reliability and highly availability.

2. MapReduce

MapReduce is principally a combination of two processes. Mapping and Reduction. Data is read from the data sources and converted to a format conducive to data analysts. This is the mapping process. The data is then subjected to mathematical operations like aggregations, grouping, encoding ranges, etc. also called as reductions and hence the terminology - reduce. In the Hadoop framework, it is the MapReduce programming model implementation for processing data on a large scale also termed as data processing at scale.

3. Hadoop Common

Hadoop common is a collection of libraries and tools required by other Hadoop modules.

4. YARN

YARN is a platform that orchestrates and manages clustered computing resources and uses them for scheduling user-applications.

5. Hadoop Ozone

Ozone is an object store (a data storage architecture that manages data as objects, through a combination of a globally unique identifier, metadata and the data itself)

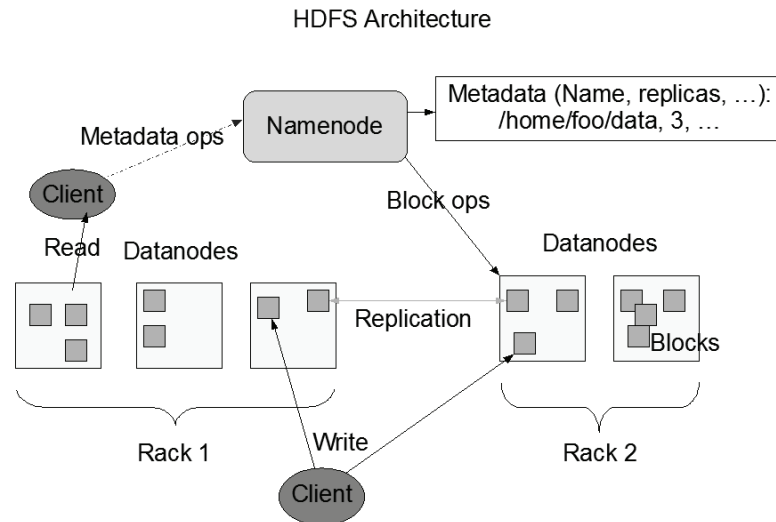


Figure 9: HDFS Architecture for Hadoop 3.3

The HDFS architecture as shown in Figure 9 implements a master/slave model and has the following components:

NameNode

It is a single node and functions as the master server managing the file-system namespace and regulating client access to files. It handles operations like opening, closing and renaming of files/directories. It also determines the mapping of blocks to DataNodes. NameNode is the arbitrator and repository for all HDFS metadata and its sole existence greatly simplifies the system architecture. The system design is such that user data never flows through the NameNode.

DataNodes

There are multiple DataNodes, usually one per node in the cluster. HDFS allows user data to be stored in files. Files are, internally, split into one or more blocks which are stored in a set of DataNodes and they handle read and write requests from the clients. Based on instruction from the NameNode, the DataNodes perform block creation, deletion and replication.

NameNodes and DataNodes are software designed to run on commodity machines. Any machine that supports Java can run the NameNode or DataNode as HDFS is built using the Java language. HDFS inherits all the advantages of Java based deployment like portability to a variety of machines.

The File System Namespace

HDFS supports hierarchical file organization and allows creation of directories with files within the directories. It supports file/directory creation, deletion, renaming or moving files/directories within the hierarchy. Additionally, it supports user quotas and access permissions including transparent encryption and snapshots.

Data Replication

HDFS reliably stores very large files across machines in a large cluster. Each file is stored as a sequence of blocks that are replicated for fault tolerance. With the exception of the last block, all blocks are uniform in size. An application can configure the number of replicas of a file. Replication factor is also configurable

before and after file creation. Files in HDFS are write-once (except for appends and truncates) and strictly have one writer at any given time. The NameNode makes all decisions pertinent to block replication and periodically receives a Blockreport and a Heartbeat from every DataNode in the cluster.

Hadoop is often used to mean both base modules and sub-modules and also the environment. It may also refer to collection of supplementary software that can be integrated with or function as extensions of Hadoop framework. Some prevalent software are Apache Flume, Apache HBase, Apache Hive, Cloudera Impala, Apache Oozie, Apache Phoenix, Apache Pig, Apache Spark, Apache Sqoop, Apache Storm and Apache ZooKeeper.

6.8.3 Conceptual Architecture of Hadoop Data Warehouse

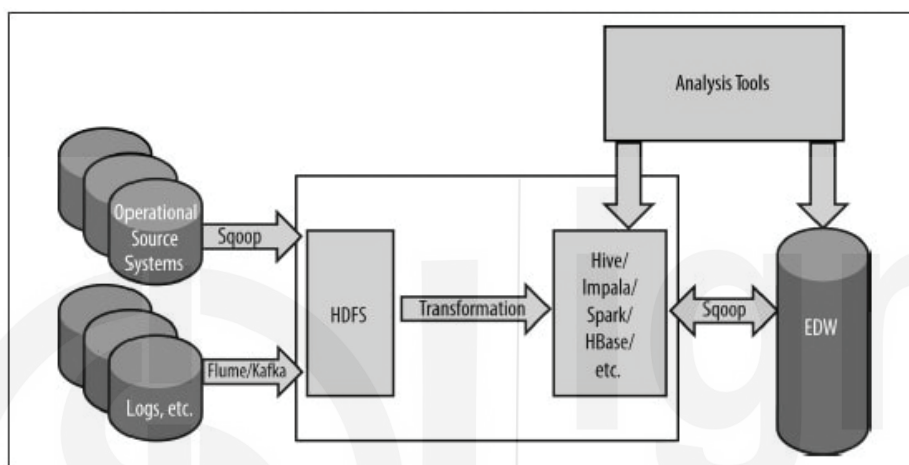


Figure 10: Conceptual Hadoop Data warehouse Architecture

A Hadoop Data warehouse can be implemented using Apache Hive or Cloudera Impala. Impala offers better performance than Hive due to its massively parallel processing (MPP) capabilities.

The architecture in Figure10 shows a Hadoop supplemented and supported DW. In this scenario, the DW inherits benefits of the Hadoop system augmenting the ETL processes that are at the core of a DW.

On the left side of the figure data sources are shown which could include other heterogeneous sources like sensor data, logs and other structured and semi-structured data. The components of DW in the context of Hadoop are explained as follows:

Data Extraction

Tools like Apache Sqoop can be utilized for data extraction from OLTP database sources. Sqoop can also be used to export data to OLTP databases after data processing is completed within the Hadoop system.

Data Transformation

Data is ingested into the Hadoop system precluding data staging. This ingested data is transformed and loaded into target data repositories within assigned directories. Tools like Spark, Cloudera Impala, Apache Hive, Apache Pig or MapReduce (the integral processing component of Hadoop) can be used for data transformations.

Data Loading

In the context of Hadoop data loading implies data hosting within Hadoop through Apache Hive or Cloudera Impala. Data can be exported to external DWs for ancillary analysis

Data Analysis / Visualization

As mentioned above data can be exported for assisting analysis through external DWs or data can be sources directly from the Hadoop system by BI, data analytics and visualization tools. Apache Sqoop can be used for the exporting data assisted through Python or Shell scripts.

6.8.4 Advantages of building a Hadoop Data Warehouse

Hadoop alleviates traditional data warehouse design challenges as follows:

- Heterogeneous data processing at scale
- Improving ETL processing throughput
- Good candidate for near-real-time data warehousing
- Efficient workload management
- Hadoop offers architectural and data level flexibility through scalable and configurable components
- The TCO of the DW system can be reduced through elastic utilization and customizations thereby, increasing the ROI

6.8.5 Challenges of building a Hadoop Data Warehouse

The challenges of a Hadoop based Data warehouse are as follows:

- Permeating the benefits of patterns in raw data to the structured components of a DW
- Data propagation latencies may arise in certain cases where large amounts of data are made available disproportionate to the data being consumed.
- Use of Hadoop as an archival repository for historical data may introduce costs for import/export of data if historical data is required.

6.9 DATA WAREHOUSE AUTOMATION

Conventional data warehouses or for that matter even some implementations of cloud data warehouses perform repetitive tasks manually with the excuse of business dynamics and constantly changing reporting parameters. This has an adverse effect on the productivity, costs and overall quality expected from the data warehousing platform.

The latest trend is towards data warehouse automation (DWA) which deals with accelerating and automating the data warehouse life-cycle (from conceptualization to implementation). Some enterprises may technically perceive it as automation beginning from analysis of source systems right up to testing and documentation processes.

6.9.1 DWA Maturity

DWA involves the use of sophisticated tools and architectural models to automate planning, design, integration and implementation of DWs through their life-cycle reducing repetitive and time consuming tasks like source analysis, ETL scripts deployment, etc. The automation process requires careful thought for evaluating and selecting the right tools that can drastically improve savings on costs and time at a fraction of TCO of data warehouses. The functionalities of automation software have matured with explosive growth of data and the race for data-driven insights to leverage businesses. Data explosion has led to the necessity for large-scale data storage and high performance integration platforms. Further, the diversity and velocity of data from various sources (like IoT, social media, etc.) require data processing at an equivalent scale.

6.9.2 Data Warehouse Automation Tools

A DWA tool provides a seamless experience, free from the hassles of coding, for integration and transition of diverse data from its source to a DW and other related components. The tool automates deployment of ETL scripts, batch-processing of data and demonstrates various functions like:

- Processes for high performance ETL and reliable ELT based integration of data
- Modeling of data at source
- Normalized, De-normalized and data structures with multiple dimensions
- Seamless integration with various data sources

Some of the essential components of DWA tools are Source Data Modeler, Dimension Modeler, Connectivity, Robust ELT engine and High Performance ETL engine

6.9.3 Advantages of DWA

A few significant advantages of DW automation are as follows:

- End-to-end data pipeline acceleration
- Automation of data capture and streaming
- Automation of data management and integration
- Optimization of data propagation paths
- Ensure automatic data flows in their entirety
- Automatic set up of target-oriented data models
- Transformation of data lakes into DWs.

Check Your Progress 5

1) How would DWA be advantageous in the case of an enterprise like Amazon?

.....

.....

.....

6.10 SUMMARY

This unit is focused on the trends in data warehousing such as Data Lakes, Complex Data Marts, Cloud Data Warehousing, Real Time Data Warehousing, Data Warehousing and Hadoop and Data Warehouse Automation. Learners are requested to keep themselves updated in this area through technical websites, newsletters etc., in order to gain knowledge in the advances made.

6.11 SOLUTIONS/ANSWERS

☞ Check your Progress 1

1. Many challenges have come to the fore-front with increasing volumes and speed of data. Data has increased due to various technological advances resulting in increase in the number of sources and diversity.

Researchers have tried to address myriad challenges subsequent to data explosion and the significant ones being worked upon by maximum researchers are as below:

- Conceptual modeling of DWs and logical data models,
 - Data warehouse loading (data-refreshing),
 - execution efficacy of OLAP queries and data mining algorithms,
 - materialized views,
 - data analysis techniques,
 - metadata management,
 - evolution management of DWs,
 - stream-based, real-time and active data warehouses, and
 - complex data warehousing (for example spatial, XML, object, multimedia)
2. The concept of logical data lakes originated through the implementation of multiple data lakes in a distributed environment. Logical data lakes form an abstraction layer over the implementation of a data lake or data lakes and allow their use without exposing the complexities of location, data sources, target systems, etc. and allow enterprises to perceive data access as an operation on a single entity that spans across multiple locations and brings data from various repositories together.
 3. A successful data lake is well cataloged both before and after data ingestion so that data traceability to its source is maintained. Data is well organized and is available through self-service for people across the enterprise. People are aware of data contained within a data lake and utilize it based on their specific needs. Other standards or best practices are adhered to all times like data governance, data quality maintenance, metadata generation, master data management, etc.

☞ Check your Progress 2

1. Complex data is data that has no definitive structure or form, for example, sensor data. Complex data also has high velocity and volumes and does not conform to any standard rules of a schema, technology or function. It is also

an attribute of Big data. Complex data forms a major portion of the raw data that is ingested into a data lake and is considered to be of significance for data insights by data engineers and data scientists.

As the race for leveraging businesses based on data-driven insights intensifies, enterprises want to derive the maximum business value hidden within the complex data. In order to do this complex data needs to be processed and converted to usable form. As conventional models are not adequately equipped to handle complex data that may be unstructured or semi-structured new models are required to efficiently process complex data and make it useful for data analysis.

☞ Check your Progress 3

1. Enterprises migrate to the cloud due to the following benefits offered by a Cloud Data Warehouse (CDW) implementation:

- a. **Improved performance**

The availability of cloud-based database management software and tools supplemented by sophisticated infrastructure allow performance that is many times faster than on-premises DWs. The ability to easily scale resources based on processing requirements and latencies are drastically reduced compared to alternative implementations of the DW.

- b. **Flexibility of cost through flexibility of cloud resources utilization**

Cloud implementations allow elastic use of resources on a pay-as-you-use basis which lends flexibility in terms of costs and resource utilization. Conversely, in on-premises based implementations the stakeholders are stuck with the resources and require additional costs to scale their infrastructure resources.

- c. **Migration of existing products to the cloud environment**

Most enterprises may have previously migrated their ERP systems or other software to the cloud or may be thinking of implementing other projects on the cloud. This makes it all the more imperative to have a CDW to reduce data propagation latencies and integration with the systems already on the cloud.

- d. **Derive benefits that are inherent to a cloud platform**

A cloud platform offers many benefits based on the myriad services it offers like Software as a service (SaaS), Platform as a service (PaaS), Infrastructure as a service (IaaS), etc. These services can be exploited to derive maximum benefits. For example by exploiting SaaS we can have best of breed software (like ETL tools, database management software, etc.) to support and supplement our Data warehouse implementation.

☞ Check your Progress 4

1. For a Real-time DW, ETL is not batch-based and uses various other approaches to load new data and update existing data, data quality needs to

be compromised as data is consumed immediately as it arrives and waiting for it to be complete in all respects (like foreign key constraints, missing values, etc.) is not permitted. Data processing practices need to be modified to process partial information and later ensure completeness through incremental updates, data integration may require partitioning to hold incomplete information. Data staging needs to be abandoned to eliminate time lost in scrutinizing data.

More you can elaborate on design, functionality, performance, data quality and data visualization.

2. CDW offers various benefits like performance, resource scalability, etc. which make it easy to implement the design changes required for Real-time DW. CDW implementation supports sophisticated tools for data storage, processing and manipulation. CDW also makes data integration easy through its toolset allowing for better data quality and data propagation across the system

☞ **Check your Progress 5**

1. Following are some of the advantages:
 - End-to-end data pipeline acceleration would increase the flow of data through its system modules (Inventory, order processing, bill-generation, delivery tracking and order fulfillment and customer feedback).
 - Automation of data capture and streaming would benefit Amazon to collect significant amount of data at various stages from when a customer buys a product to order fulfillment and feedback
 - Automation of data management and integration would aid Amazon to avoid data duplication, reduce the risks and challenges of data import / export to from other systems, generate data value for business and increase QoS.
 - Optimization of data propagation paths will ensure data availability with minimum effort and maximum benefit
 - Amazon can ensure automatic data flows in their entirety i.e. ensure that all the details required for a transaction are completely captured and the data is automatically delivered to the systems requesting it for analysis.
 - Automatic set up of target-oriented data models would enable Amazon to ensure that data reaches its intended audience through the automatic selection of data source, data processing and integration tools and finally the analytics systems.
 - As Amazon operates on a very large scale spanning various continents the amount of data generated would be humongous and contained in data lakes. Transformation of data lakes into DWs would enable Amazon to ensure regulatory compliances. It would also allow data localization in the case of countries requiring it for political and other reasons. Through all this the DWs can still be integrated and enjoy the benefits of a data lake ecosystem.

6.12 FURTHER READINGS

1. Robert Wrembel, Alberto Abelló and Il-Yeol Song, *DOLAP data warehouse research over two decades: Trends and challenges. 2019*, Information Systems, pp. 44-47, Elsevier.
2. Bill Inmon, ‘*Data Lake Architecture: Designing the Data Lake and Avoiding the Garbage Dump*’, 2016, ISBN- 9781634621205, Technics Publications (<https://www.technicspub.com/>)
3. Alex Gorelik, ‘*The Enterprise Big Data Lake: Delivering the Promise of Big Data and Data Science*’, 2019, ISBN- 9781491931554, O’Reilly Media Inc.
4. Vassiliadis, P., Simitsis, A.: Near Real Time ETL, pp. 1-31. Springer US, Boston, MA (2009), http://dx.doi.org/10.1007/978-0-387-87431-9_2
5. Obali, M., Dursun, B., Erdem, Z., Grr, A .K.: *A real time data warehouse approach for data processing. In: Signal Processing and Communications Applications Conference SIU, 2013* 21st. pp. 1-4 (2013)
6. Kimball, Ralph, Ross Margy.: ‘*The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*’, Third Edition, 2013, John Wiley & Sons Inc.

