(Big) Data Engineering In Depth

From Beginner to Professional

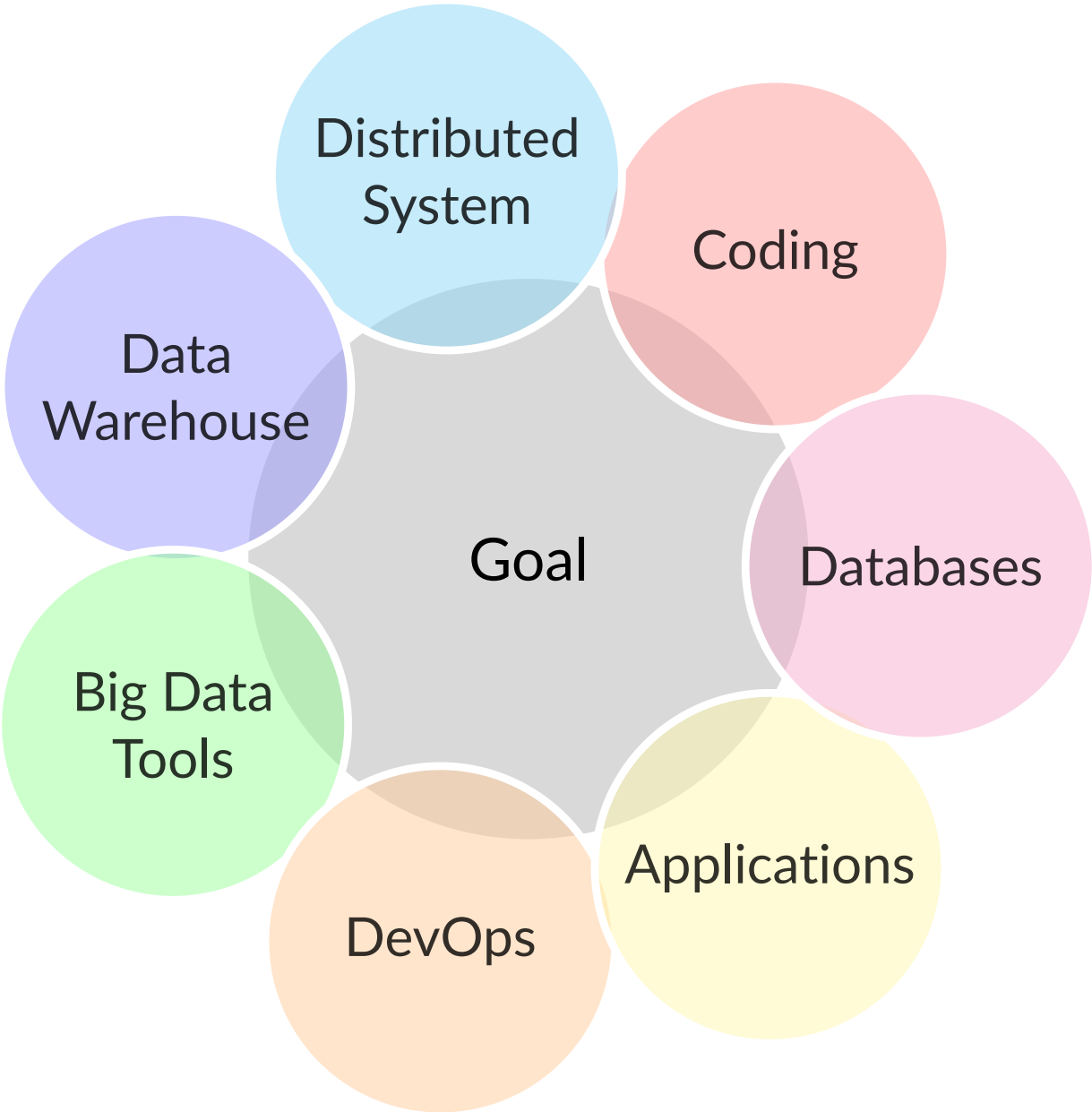
Mostafa Alaa Mohamed
Senior Big Data Engineer
MoustafaAlaa Moustafa Alaa @Moustafa_alaa22
mustafa.alaa.mohamed@gmail.com

[1]Big Data & Analytics Department, Epam Systems

The Definitive Guide to Big Data Engineering Tasks

Course Introduction

Course Target



Learning Objectives

- Understand the data management life-cycle.

Learning Objectives

- Understand the data management life-cycle.
- Illustrate the basics of distributed systems concepts

Learning Objectives

- Understand the data management life-cycle.
- Illustrate the basics of distributed systems concepts
- Be familiar with ETL for (batch/steaming) data over distributed systems ex: Hadoop & Spark.

Learning Objectives

- Understand the data management life-cycle.
- Illustrate the basics of distributed systems concepts
- Be familiar with ETL for (batch/steaming) data over distributed systems ex: Hadoop & Spark.
- Apply QA and testing for the data pipeline cycle.

Learning Objectives

- Understand the data management life-cycle.
- Illustrate the basics of distributed systems concepts
- Be familiar with ETL for (batch/steaming) data over distributed systems ex: Hadoop & Spark.
- Apply QA and testing for the data pipeline cycle.
- Automate the data life-cycle process end-to-end (e2e).

Learning Objectives

- Understand the data management life-cycle.
- Illustrate the basics of distributed systems concepts
- Be familiar with ETL for (batch/steaming) data over distributed systems ex: Hadoop & Spark.
- Apply QA and testing for the data pipeline cycle.
- Automate the data life-cycle process end-to-end (e2e).
- Building real-life examples.

Learning Objectives

- Understand the data management life-cycle.
- Illustrate the basics of distributed systems concepts
- Be familiar with ETL for (batch/steaming) data over distributed systems ex: Hadoop & Spark.
- Apply QA and testing for the data pipeline cycle.
- Automate the data life-cycle process end-to-end (e2e).
- Building real-life examples.
- Applying machine learning over big data.

Learning Objectives

- Understand the data management life-cycle.
- Illustrate the basics of distributed systems concepts
- Be familiar with ETL for (batch/steaming) data over distributed systems ex: Hadoop & Spark.
- Apply QA and testing for the data pipeline cycle.
- Automate the data life-cycle process end-to-end (e2e).
- Building real-life examples.
- Applying machine learning over big data.
- Understanding of the DevOps tools and functions in data life-cycle.

Learning Objectives

- Understand the data management life-cycle.
- Illustrate the basics of distributed systems concepts
- Be familiar with ETL for (batch/steaming) data over distributed systems ex: Hadoop & Spark.
- Apply QA and testing for the data pipeline cycle.
- Automate the data life-cycle process end-to-end (e2e).
- Building real-life examples.
- Applying machine learning over big data.
- Understanding of the DevOps tools and functions in data life-cycle.
- Build and scale your data product.

Learning Objectives

- Understand the data management life-cycle.
- Illustrate the basics of distributed systems concepts
- Be familiar with ETL for (batch/steaming) data over distributed systems ex: Hadoop & Spark.
- Apply QA and testing for the data pipeline cycle.
- Automate the data life-cycle process end-to-end (e2e).
- Building real-life examples.
- Applying machine learning over big data.
- Understanding of the DevOps tools and functions in data life-cycle.
- Build and scale your data product.
- Simplify the concepts in data management.

Audience: Who Should Take This Course?

- Data Engineer who needs to get more knowledge in distributed systems and Big Data.

Audience: Who Should Take This Course?

- Data Engineer who needs to get more knowledge in distributed systems and Big Data.
- Data Warehouse Engineer who needs to know more about big data.

Audience: Who Should Take This Course?

- Data Engineer who needs to get more knowledge in distributed systems and Big Data.
- Data Warehouse Engineer who needs to know more about big data.
- Java developer who needs to change to data engineering track.

Audience: Who Should Take This Course?

- Data Engineer who needs to get more knowledge in distributed systems and Big Data.
- Data Warehouse Engineer who needs to know more about big data.
- Java developer who needs to change to data engineering track.
- DevOps engineers who needs to understand the concepts of big data.

Audience: Who Should Take This Course?

- Data Engineer who needs to get more knowledge in distributed systems and Big Data.
- Data Warehouse Engineer who needs to know more about big data.
- Java developer who needs to change to data engineering track.
- DevOps engineers who needs to understand the concepts of big data.
- Business or entrepreneur who needs to get more information about how to build or manage a data product.

Audience: Who Should Take This Course?

- Data Engineer who needs to get more knowledge in distributed systems and Big Data.
- Data Warehouse Engineer who needs to know more about big data.
- Java developer who needs to change to data engineering track.
- DevOps engineers who needs to understand the concepts of big data.
- Business or entrepreneur who needs to get more information about how to build or manage a data product.

Getting max benefit from this course

Take the course advantage

- Follow the videos order as described.

Getting max benefit from this course

Take the course advantage

- Follow the videos order as described.
- Read the references for each section (including the implementation of the examples if exists).

Getting max benefit from this course

Take the course advantage

- Follow the videos order as described.
- Read the references for each section (including the implementation of the examples if exists).
- Repeat the lecture code with your own.

Getting max benefit from this course

Take the course advantage

- Follow the videos order as described.
- Read the references for each section (including the implementation of the examples if exists).
- Repeat the lecture code with your own.
- Do the assignments.

Getting max benefit from this course

Take the course advantage

- Follow the videos order as described.
- Read the references for each section (including the implementation of the examples if exists).
- Repeat the lecture code with your own.
- Do the assignments.
- Ask your questions.

Getting max benefit from this course

Take the course advantage

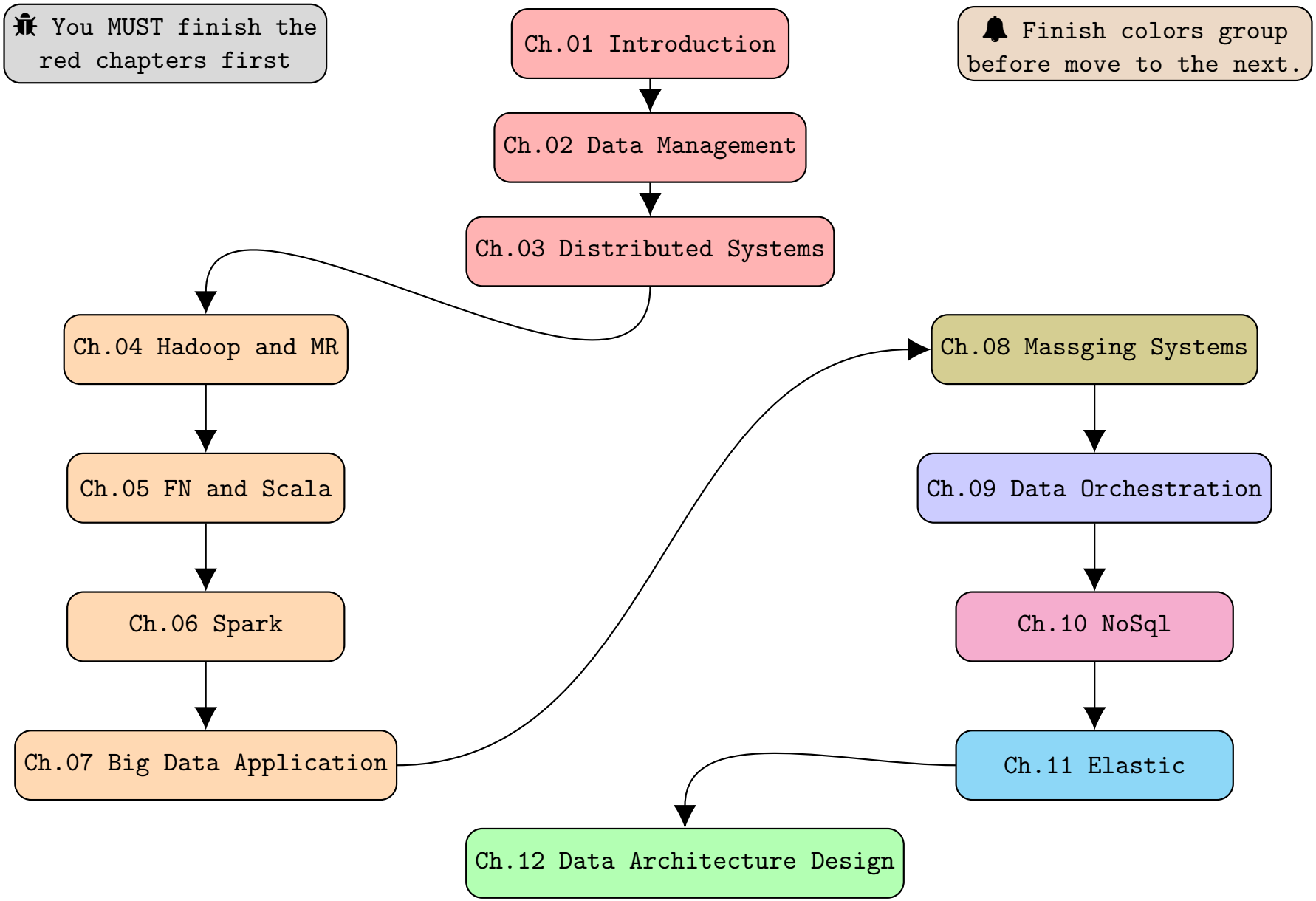
- Follow the videos order as described.
- Read the references for each section (including the implementation of the examples if exists).
- Repeat the lecture code with your own.
- Do the assignments.
- Ask your questions.
- Join the online meeting or discussions.

Getting max benefit from this course

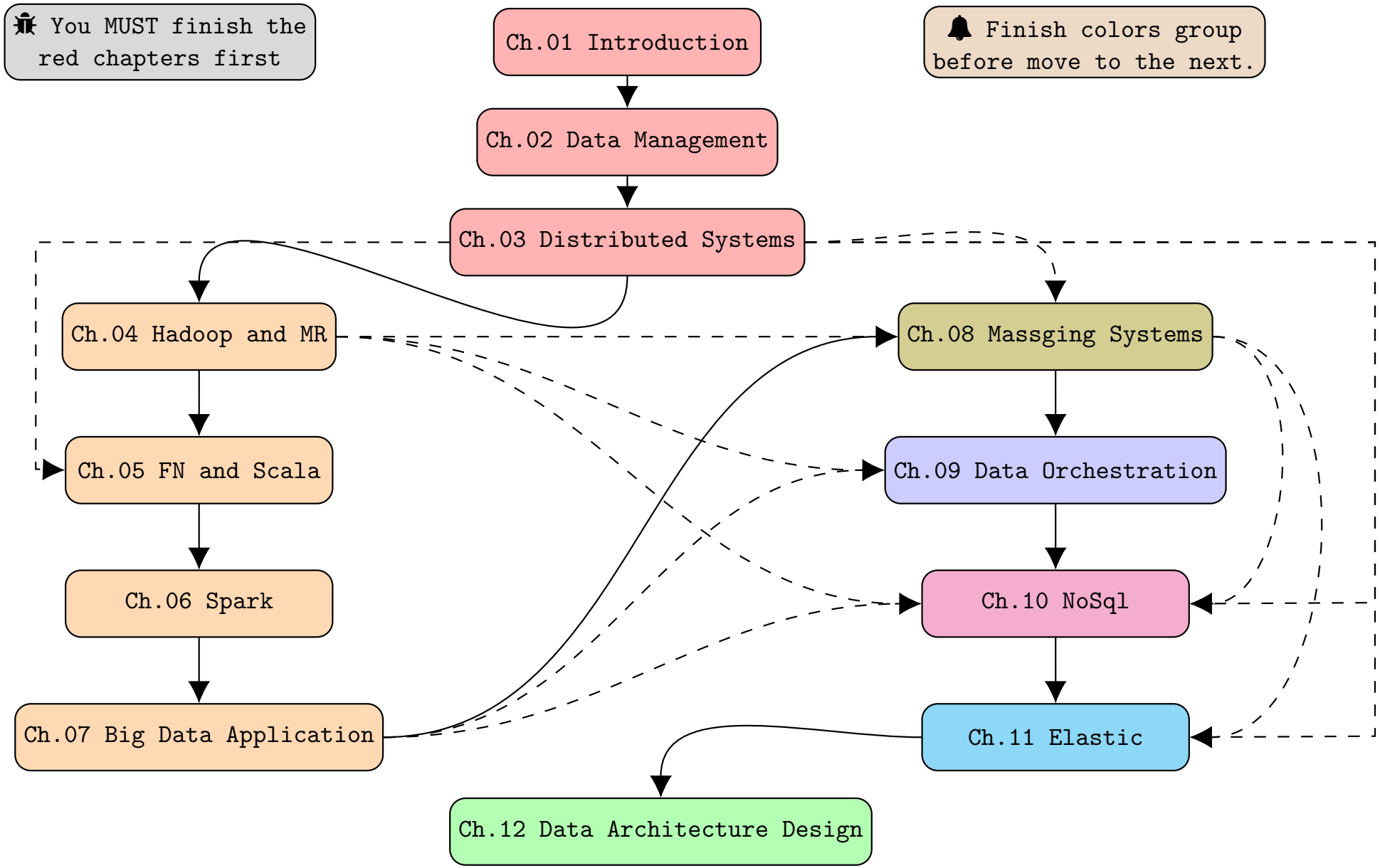
Take the course advantage

- Follow the videos order as described.
- Read the references for each section (including the implementation of the examples if exists).
- Repeat the lecture code with your own.
- Do the assignments.
- Ask your questions.
- Join the online meeting or discussions.

Chapter Dependencies



Chapter Dependencies (Jump Out Path)



Assignments and Labs

Remark

- Full project code.

Assignments and Labs

Remark

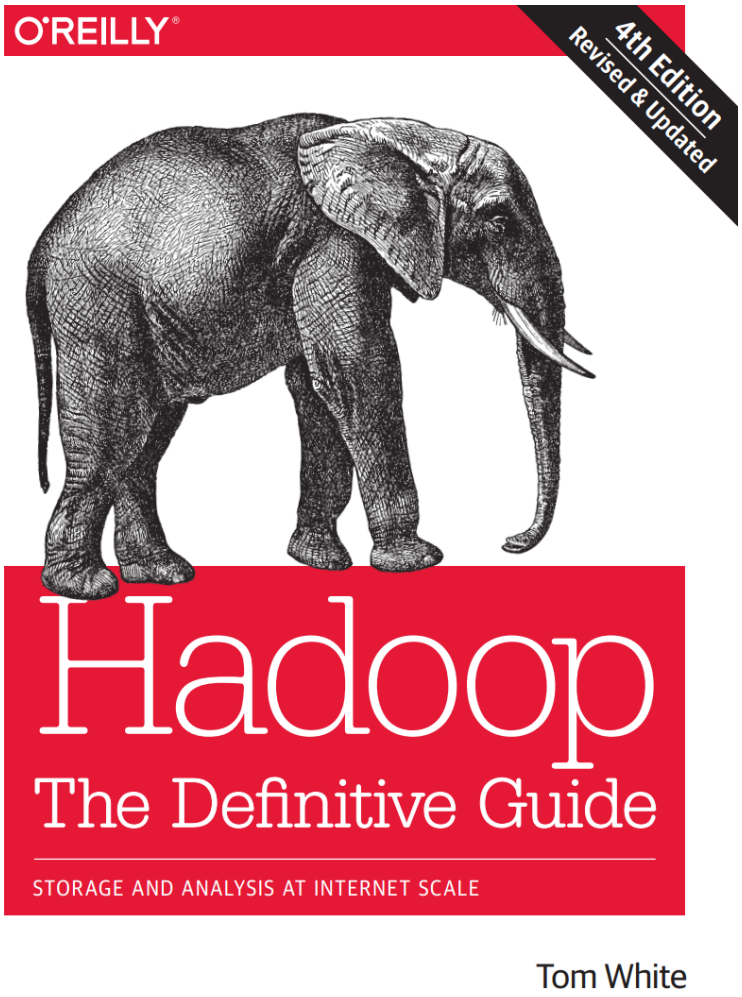
- Full project code.
- Notebooks (Jupyter or Zeppelin).

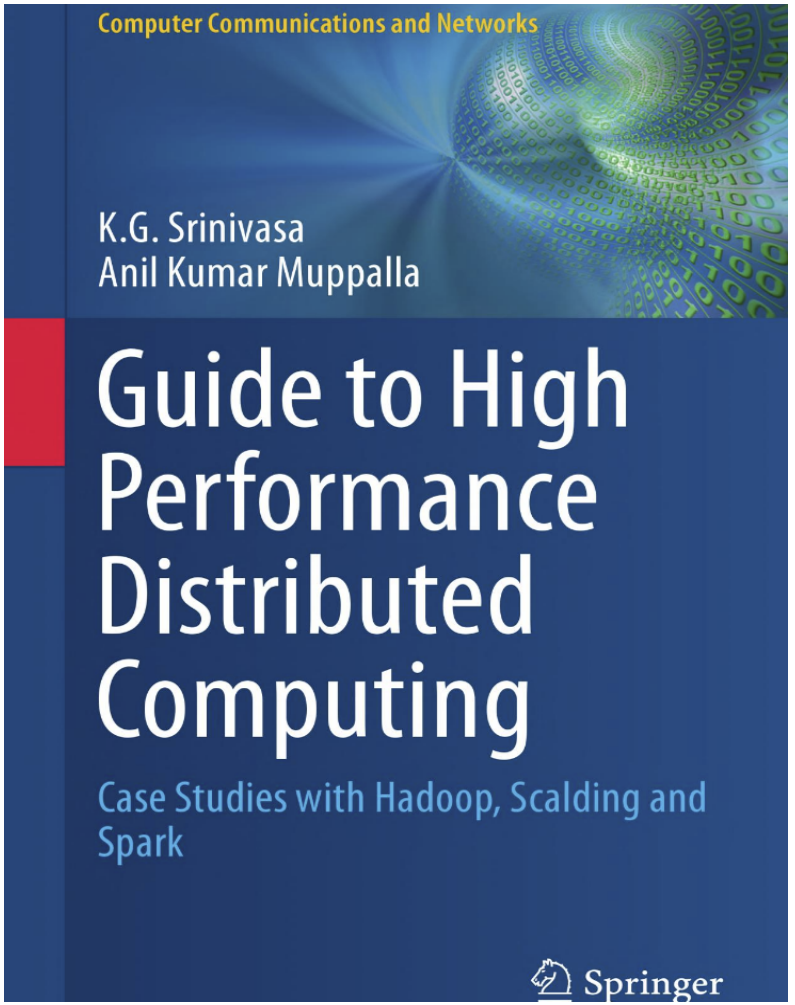
Assignments and Labs

Remark

- Full project code.
- Notebooks (Jupyter or Zeppelin).
- Read the reference.

Textbooks-1

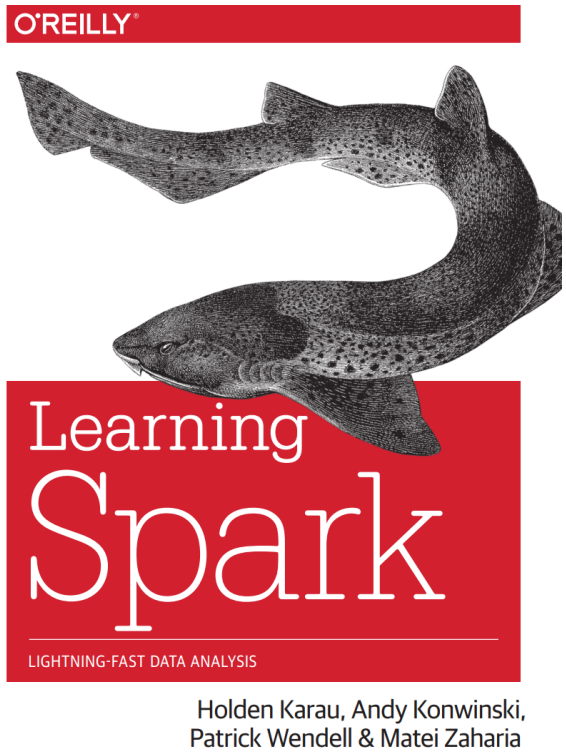

O'REILLY
4th Edition
Revised & Updated
Hadoop
The Definitive Guide
STORAGE AND ANALYSIS AT INTERNET SCALE
Tom White


Computer Communications and Networks
K.G. Srinivasa
Anil Kumar Muppalla
Guide to High Performance Distributed Computing
Case Studies with Hadoop, Scalding and Spark
Springer

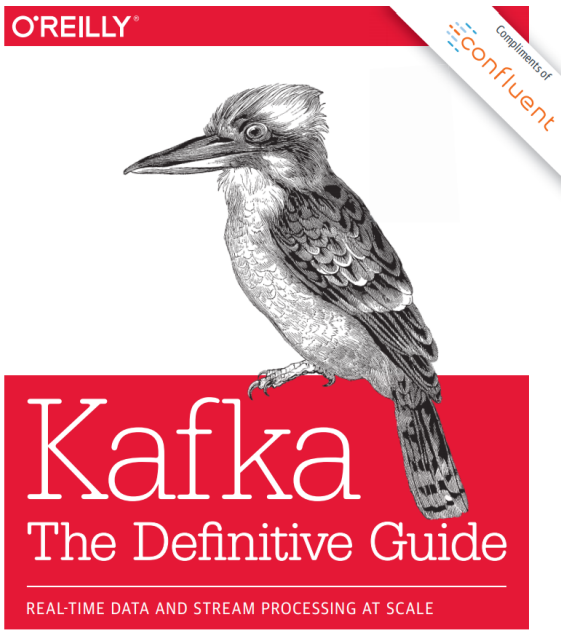
Textbooks-2

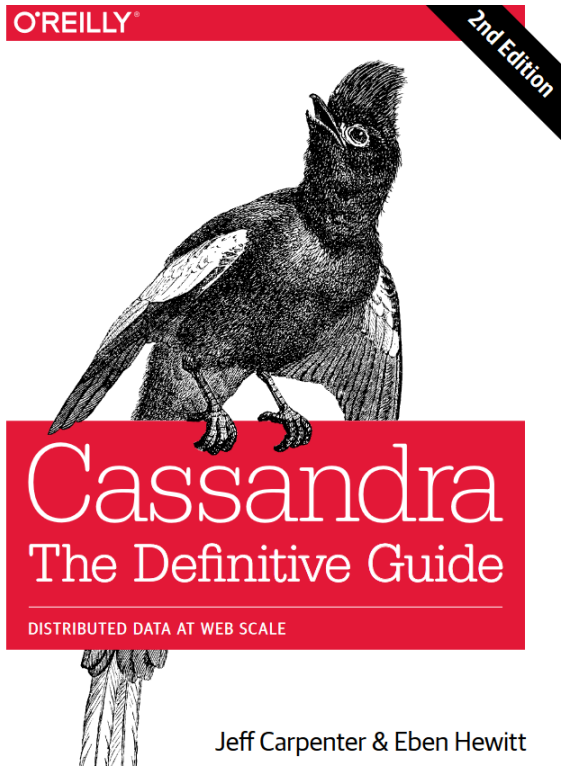

CATEGORY THEORY
FOR PROGRAMMERS
Bartosz Milewski

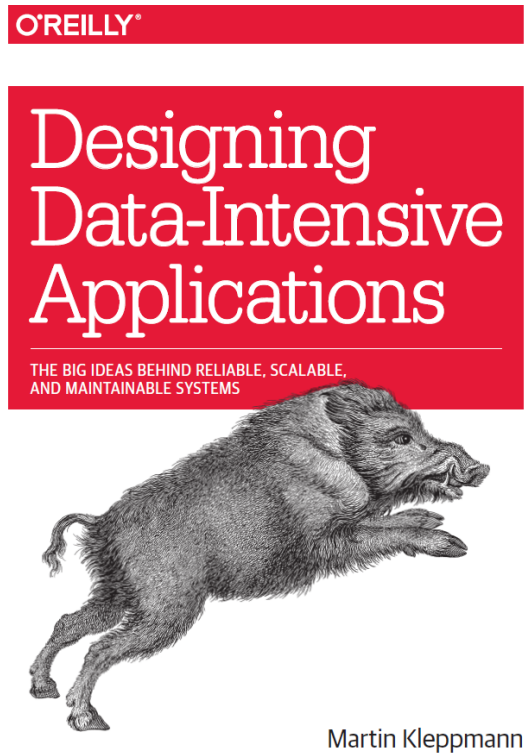

O'REILLY
High Performance
Spark
BEST PRACTICES FOR SCALING
& OPTIMIZING APACHE SPARK
Holden Karau &
Rachel Warren


O'REILLY
Learning
Spark
LIGHTNING-FAST DATA ANALYSIS
Holden Karau, Andy Konwinski,
Patrick Wendell & Matei Zaharia

Textbooks-3


O'REILLY
Compliments of
confluent
Kafka
The Definitive Guide
REAL-TIME DATA AND STREAM PROCESSING AT SCALE
Neha Narkhede,
Gwen Shapira & Todd Palino


O'REILLY
2nd Edition
Cassandra
The Definitive Guide
DISTRIBUTED DATA AT WEB SCALE
Jeff Carpenter & Eben Hewitt


O'REILLY
Designing
Data-Intensive
Applications
THE BIG IDEAS BEHIND RELIABLE, SCALABLE,
AND MAINTAINABLE SYSTEMS
Martin Kleppmann

Ugly but important

- User stories or technical discussions are not related to any of my current work or my previous companies.
- I am working at EPAM Systems. My company approved me for doing this online course public but the materials are not reviewed or assessed by my company. It is on my own responsibilities.

Table of Contents I

Table of Contents II

Table of Contents III

Table of Contents IV

Table of Contents V

Table of Contents VI

Table of Contents VII

Introduction To Data Management and Data Warehouse

Chapter Objectives

- Be familiar with data management life-cycle.

Chapter Objectives

- Be familiar with data management life-cycle.
- Introduction to data warehouse and its usage.

Chapter Objectives

- Be familiar with data management life-cycle.
- Introduction to data warehouse and its usage.
- Motivation to DWH.

Chapter Objectives

- Be familiar with data management life-cycle.
- Introduction to data warehouse and its usage.
- Motivation to DWH.
- What is the different types of DWH?

Chapter Objectives

- Be familiar with data management life-cycle.
- Introduction to data warehouse and its usage.
- Motivation to DWH.
- What is the different types of DWH?
- Use cases

Chapter Objectives

- Be familiar with data management life-cycle.
- Introduction to data warehouse and its usage.
- Motivation to DWH.
- What is the different types of DWH?
- Use cases
- Data Encoding and Formats

Chapter Objectives

- Be familiar with data management life-cycle.
- Introduction to data warehouse and its usage.
- Motivation to DWH.
- What is the different types of DWH?
- Use cases
- Data Encoding and Formats
- Challenges to build a DWH.

Chapter Objectives

- Be familiar with data management life-cycle.
- Introduction to data warehouse and its usage.
- Motivation to DWH.
- What is the different types of DWH?
- Use cases
- Data Encoding and Formats
- Challenges to build a DWH.
- Data modeling design.

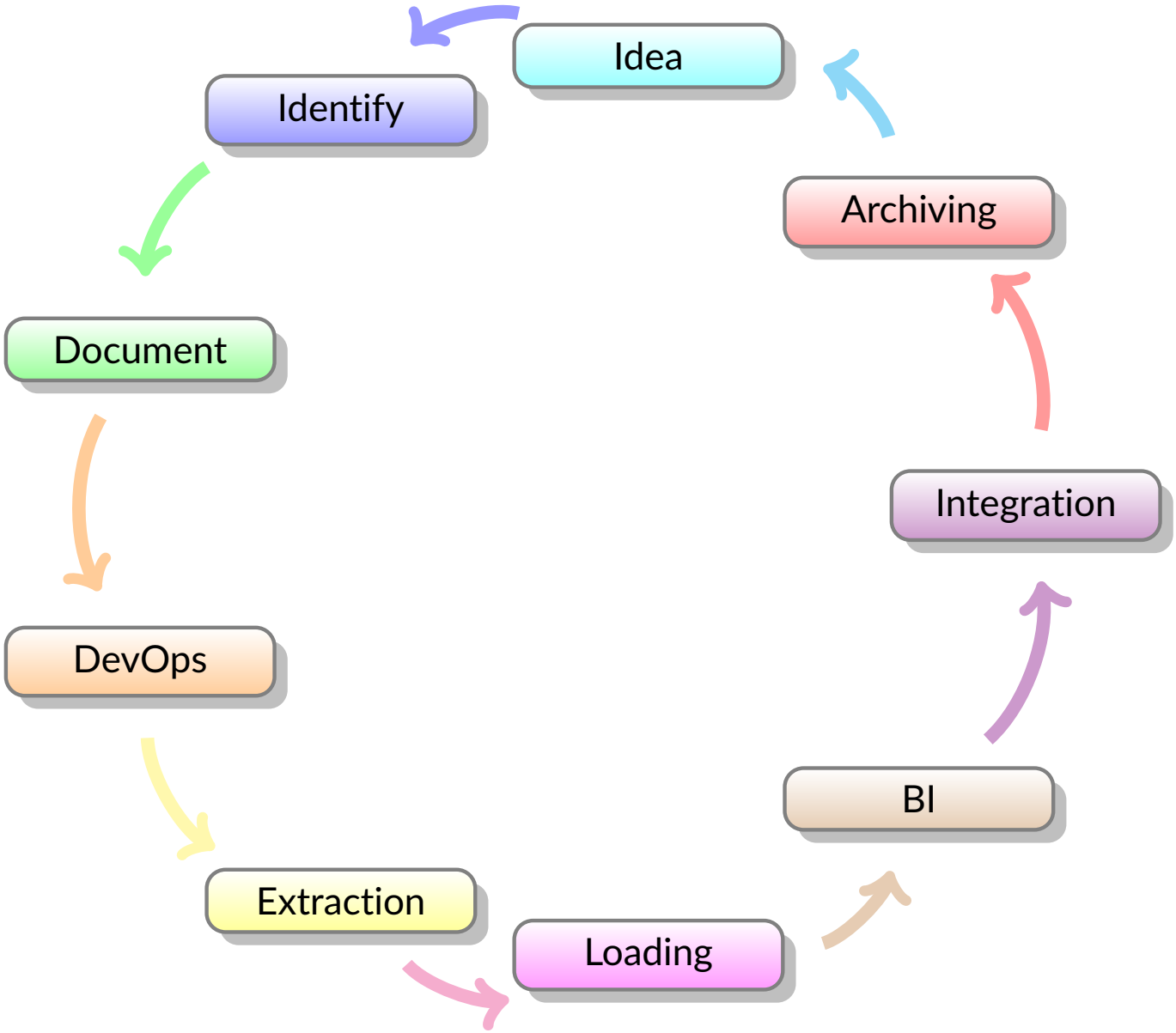
Chapter Objectives

- Be familiar with data management life-cycle.
- Introduction to data warehouse and its usage.
- Motivation to DWH.
- What is the different types of DWH?
- Use cases
- Data Encoding and Formats
- Challenges to build a DWH.
- Data modeling design.

Data Management

- Data are a product.
- Data product has a life-cycle as following (simplified):
  - **Question**, Idea, or service.
  - **Identifying** the source of information and the data type ex: (text, images, videos, audio, or sensors).
  - **Document** all details regarding the data including quality, security, efficiency, and access (consideration during the cycle).
  - Delivery automation (Tools and Process) AKA **DevOps** cycle.
  - **Extraction** Process (collection).
  - **Transformation** ex: (cleansing, Apply business logic, Organize).
  - **Loading** or store the transformed data based on our usage or use case.
  - Business Intelligence (**BI**) or data discovery (continues process).
  - **Integration** and publishing.
  - Data retention or **archiving** process ex: (Hot or Cold storage).

Data Management Life-Cycle



Motivation to Data Warehouse

- Data could be a product for some companies.

Motivation to Data Warehouse

- Data could be a product for some companies.
- It could be decision support for other products or businesses.

Motivation to Data Warehouse

- Data could be a product for some companies.
- It could be decision support for other products or businesses.
- Reporting the results after pass the data life-cycle will be from storage (Database).

Motivation to Data Warehouse

- Data could be a product for some companies.
- It could be decision support for other products or businesses.
- Reporting the results after pass the data life-cycle will be from storage (Database).
- There are some challenges facing the people who work on data management backend:

Motivation to Data Warehouse

- Data could be a product for some companies.
- It could be decision support for other products or businesses.
- Reporting the results after pass the data life-cycle will be from storage (Database).
- There are some challenges facing the people who work on data management backend:
  - Performance.

Motivation to Data Warehouse

- Data could be a product for some companies.
- It could be decision support for other products or businesses.
- Reporting the results after pass the data life-cycle will be from storage (Database).
- There are some challenges facing the people who work on data management backend:
  - Performance.
  - Integration.

Motivation to Data Warehouse

- Data could be a product for some companies.
- It could be decision support for other products or businesses.
- Reporting the results after pass the data life-cycle will be from storage (Database).
- There are some challenges facing the people who work on data management backend:
  - Performance.
  - Integration.
  - Applying analytical functions.

Motivation to Data Warehouse

- Data could be a product for some companies.
- It could be decision support for other products or businesses.
- Reporting the results after pass the data life-cycle will be from storage (Database).
- There are some challenges facing the people who work on data management backend:
  - Performance.
  - Integration.
  - Applying analytical functions.
- Vendors who are working to solve the above challenges creating their own product of DWH and their ultimate work is to optimize the above points.

Motivation to Data Warehouse (DWH)

- The DWH is not a product but an environment.
- It is a process of transforming data into information and make it available to users in a **timely manner** to make a difference.
- It is an architectural construct of an information system which provides users with current and historical decision support information which is difficult to access or present in the traditional operational data store.
- The DWH is the core of the BI system which is built for data analysis and reporting.

Definition (What is Data Warehousing?)

A DWH is defined as a technique for collecting and managing data from varied sources to **provide meaningful business insights**. It is a blend of technologies and components which aids the strategic use of data.

Motivation to Data Warehouse

Data warehouse system is also known by the following names:

- Decision Support System (DSS).
- Business Intelligence Solution.
- Executive Information System.
- Management Information System.
- Analytic Application.
- Data Warehouse.

The real concept was given by Inmon Bill. He was considered as a father of the DWH. He had written about a variety of topics for building, usage, and maintenance of the warehouse & the Corporate Information Factory

Motivation to Data Warehouse

Types of Data Warehouse

Enterprise Data Warehouse (EDWH) It provides decision support service across the enterprise. It offers a unified approach for organizing and representing data (DWH Model). It offers data classifications according to the subject with privileges policy.

Operational Data Store (ODS): is a central database that provides an up-to-date (real-time) data from multiple transnational systems for operational reporting into a single DWH.

Data Mart: A data mart is a subset of the data warehouse. It specially designed for a particular line of business, such as sales, finance, sales or finance. In an independent data mart, data can collect directly from sources.

DWH vs ODS vs Data Mart

Metric	DWH	ODS	Data Mart
Latency	Day -1	Real-time	Day -1
Data level	Transnational	Transnational	Summary
Historical	Long-term	Snapshot	Aggregated Long-Term
Size	TB/PB	GB	GB/TB
Orientation	Multi sources	Multi sources	Product
Business Units	Multi organizational units	Product team	Business team

DWH vs Operational databases

Metric	Transactions DB	DWH
Volume	GB/TB	TB/PB
Historical	Short-term	Long-Term
rows	¡1000M	1000M¿
Orientation	Product	Subject or multi products
Business Units	Product team	Multi organizational units
Normalization	Normalized	Not required (De-normalized in many use cases)
Data Model	Relational	Star Schema or Multi-dim
Intelligence	Reporting	Advanced reporting and Machine Learning
Use cases	Online transactions & operations	Centeralized storage (360°)

Transnational DB Use cases

Transnational DB Use cases

DWH Use cases

DWH Use cases

DWH Use cases

Use case (Operational DB)

- A telecommunication company named **XTec**.

Use case (Operational DB)

- A telecommunication company named **XTec**.
- They have lots of systems. One of this systems is a CRM system as example of operational DB.

Use case (Operational DB)

- A telecommunication company named **XTec**.
- They have lots of systems. One of this systems is a CRM system as example of operational DB.
  - The CRM system handles the customer activities with the company including (sales, change in customer plans, and other activities).

Use case (Operational DB)

- A telecommunication company named **XTec**.
- They have lots of systems. One of this systems is a CRM system as example of operational DB.
  - The CRM system handles the customer activities with the company including (sales, change in customer plans, and other activities).
  - This system has a backend database (MySQL).

Use case (Operational DB)

- A telecommunication company named **XTec**.
- They have lots of systems. One of this systems is a CRM system as example of operational DB.
  - The CRM system handles the customer activities with the company including (sales, change in customer plans, and other activities).
  - This system has a backend database (MySQL).
  - CRM team can report their sales and customer activities from their database.

Use case (Operational DB)

- A telecommunication company named **XTec**.
- They have lots of systems. One of this systems is a CRM system as example of operational DB.
  - The CRM system handles the customer activities with the company including (sales, change in customer plans, and other activities).
  - This system has a backend database (MySQL).
  - CRM team can report their sales and customer activities from their database.
  - Product owner can take a decision based on their system backend reports.

Use case (DWH)

- What is the need for DWH?

Use case (DWH)

- What is the need for DWH?
  - This company has other systems for example: billing, charging, signaling.

Use case (DWH)

- What is the need for DWH?
  - This company has other systems for example: billing, charging, signaling.
  - They need to report information related to the CRM, billing, and signaling source systems in one report.

Use case (DWH)

- What is the need for DWH?
  - This company has other systems for example: billing, charging, signaling.
  - They need to report information related to the CRM, billing, and signaling source systems in one report.
  - So, they need to ingest (transfer) the data from the source systems to one single database.

Use case (DWH)

- What is the need for DWH?
  - This company has other systems for example: billing, charging, signaling.
  - They need to report information related to the CRM, billing, and signaling source systems in one report.
  - So, they need to ingest (transfer) the data from the source systems to one single database.
  - The decision from the DHW is a **global and strategical decision.**

Use case (DWH)

- What is the need for DWH?
  - This company has other systems for example: billing, charging, signaling.
  - They need to report information related to the CRM, billing, and signaling source systems in one report.
  - So, they need to ingest (transfer) the data from the source systems to one single database.
  - The decision from the DHW is a **global and strategical decision.**
  - If the company needs to build a machine learning model which needs data from different sources. They need to load the data from a centralized database rather than read each source alone.

Use case (DWH)

The Full picture required a DWH. However, we still need the other operational databases for product development perspective.

Use case (ODS)

- Why do we need the ODS?

Use case (ODS)

- Why do we need the ODS?
- How does it fit in our system?

Use case (ODS)

XTec has a call center system which handles the customer inquiries. This system requires the some data related to usage, customer information, billing details to be calculated and accumulated in **real-time** to be able to give the customer the right answer for his inquires.

Use case (ODS)

- So, What is the challenge for this system?

Use case (ODS)

- So, What is the challenge for this system?
  - It needs specific information from different source systems.

Use case (ODS)

- So, What is the challenge for this system?
  - It needs specific information from different source systems.
  - It requires to track the source system database changes or update in real-time.

Use case (ODS)

- So, What is the challenge for this system?
  - It needs specific information from different source systems.
  - It requires to track the source system database changes or update in real-time.
  - It's functionality is based on the aggregate data not the transactions for example (It needs the total outgoing calls till time or it needs the total charging amounts from prepaid or the available limits from billing if it is postpaid).

Use case (ODS)

- ODS is based on change data capture (CDC). This approach used to determine the data change and apply action based on this change.

Use case (ODS)

- ODS is based on change data capture (CDC). This approach used to determine the data change and apply action based on this change.
- ODS uses the real-time aggregations to support the online systems from different source systems.

DWH Architecture Overview

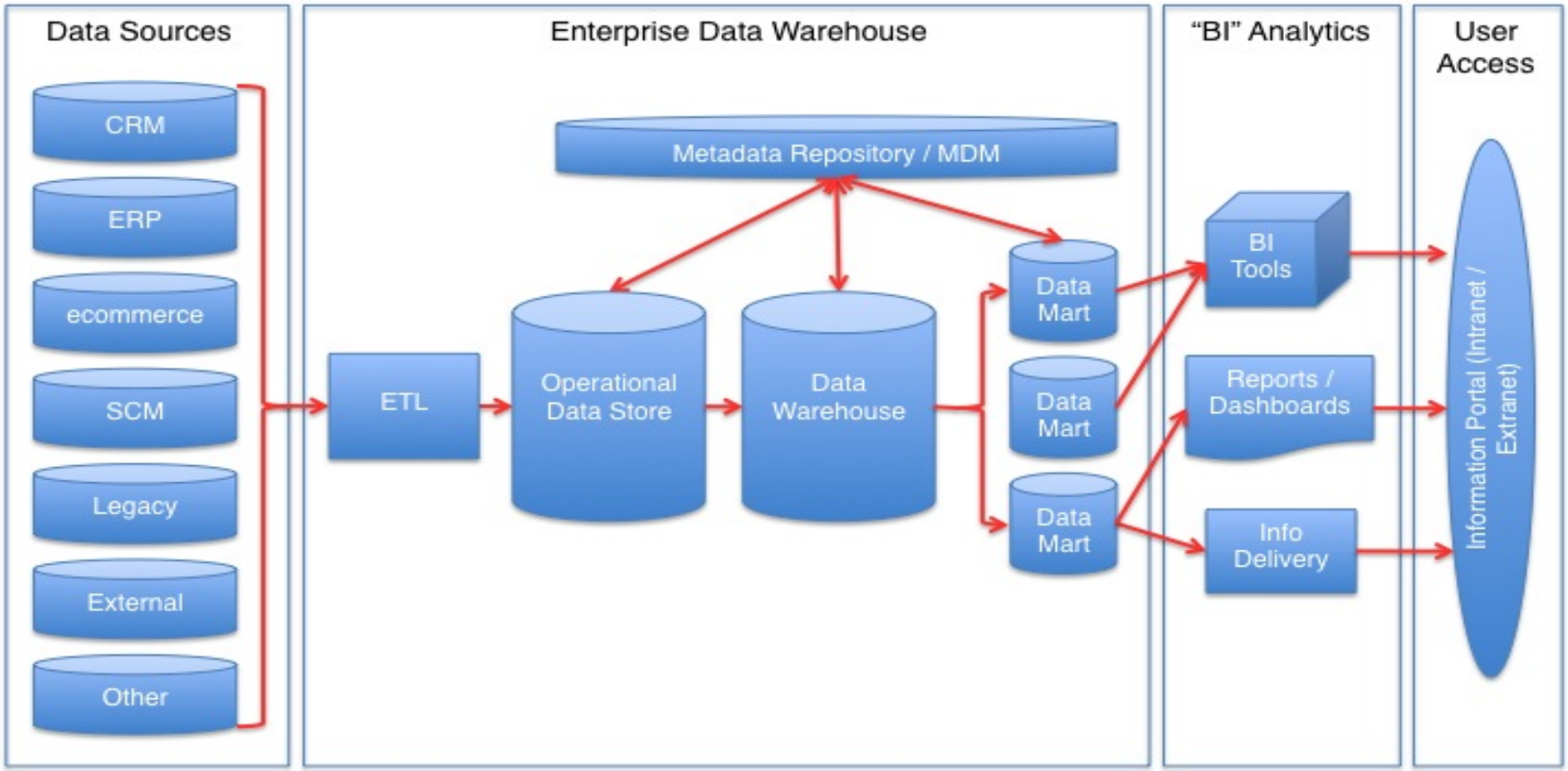


Figure: taken from XXXX

Data Abstraction

Database systems are made-up of complex data structures. To ease the user interaction with database, the developers hide internal irrelevant details from users. This process of hiding irrelevant details from user is called data abstraction.

- There are 3 levels of data abstraction.

Data Abstraction

Database systems are made-up of complex data structures. To ease the user interaction with database, the developers hide internal irrelevant details from users. This process of hiding irrelevant details from user is called data abstraction.

- There are 3 levels of data abstraction.
  - Physical level

Data Abstraction

Database systems are made-up of complex data structures. To ease the user interaction with database, the developers hide internal irrelevant details from users. This process of hiding irrelevant details from user is called data abstraction.

- There are 3 levels of data abstraction.
  - Physical level
  - Logical/ Conceptual level.

Data Abstraction

Database systems are made-up of complex data structures. To ease the user interaction with database, the developers hide internal irrelevant details from users. This process of hiding irrelevant details from user is called data abstraction.

- There are 3 levels of data abstraction.
  - Physical level
  - Logical/ Conceptual level.
  - View level.

Data Abstraction

- **Physical level**: This is the lowest level of data abstraction. It describes how data is actually stored in database. You can get the complex data structure details at this level.

Data Abstraction

- **Physical level**: This is the lowest level of data abstraction. It describes how data is actually stored in database. You can get the complex data structure details at this level.
- **Logical level**: This is the middle level of 3-level data abstraction architecture. It describes what data is stored in database.

Data Abstraction

- **Physical level**: This is the lowest level of data abstraction. It describes how data is actually stored in database. You can get the complex data structure details at this level.
- **Logical level**: This is the middle level of 3-level data abstraction architecture. It describes what data is stored in database.
- **View level**: Highest level of data abstraction. This level describes the user interaction with database system.

Data Abstraction

Example: Let's say we are storing customer information in a customer table. At physical level these records can be described as blocks of storage (bytes, gigabytes, terabytes etc.) in memory. These details are often hidden from the programmers.
At the logical level these records can be described as fields and attributes along with their data types, their relationship among each other can be logically implemented. The programmers generally work at this level because they are aware of such things about database systems.
At view level, user just interact with system with the help of GUI and enter the details at the screen, they are not aware of how the data is stored and what data is stored; such details are hidden from them. **Data Models is the logical level.** we will discribte the logical level and how can we propsose the external view for the end users. Regarding the physical level we will not dig dive at this level but for the next chapters we will discuss it in Hadoop, Kafka, Cassandra.

What is data model?

Data model is

- An abstract model that organizes elements of data.

What is data model?

Data model is

- An abstract model that organizes elements of data.
- It describes the objects, entities and data structure properties, semantic, and constraint.

What is data model?

Data model is

- An abstract model that organizes elements of data.
- It describes the objects, entities and data structure properties, semantic, and constraint.
- It formalizes the relationship between entities.

What is data model?

Data model is

- An abstract model that organizes elements of data.
- It describes the objects, entities and data structure properties, semantic, and constraint.
- It formalizes the relationship between entities.
- It describes how application (report) API data manipulation.

What is data model?

Data model is

- An abstract model that organizes elements of data.
- It describes the objects, entities and data structure properties, semantic, and constraint.
- It formalizes the relationship between entities.
- It describes how application (report) API data manipulation.
- It describes the conceptual design of a business or an application with its flow, logic, semantic information (rules), and how things are done.

What is data model?

Data model is

- An abstract model that organizes elements of data.
- It describes the objects, entities and data structure properties, semantic, and constraint.
- It formalizes the relationship between entities.
- It describes how application (report) API data manipulation.
- It describes the conceptual design of a business or an application with its flow, logic, semantic information (rules), and how things are done.
- It refers to a set of concepts used in defining such as entities, attributes, relations, or tables.

What is data model?

Data model is not

- a science.
- a static design for each organization.
- a type of database.
- a new invention which needs to be done for each project.

Data model is

- an engineering design practices.
- a general concepts which lead to build full architecture.
- different based on the use case and the database type.
- customizable and we can utilize some of ready built architecture.
- implementing using different ways.
- affecting the information reporting performance and ways.

Why does data models are important?

- Data models are currently affecting software design.
- It decides how engineers will think about the problem they are solving.

Data Model Design vs Implementation

- You need to build a home. So, how do we design this home?

Data Model Design vs Implementation

- You need to build a home. So, how do we design this home?
  - Determine if the home is one level or multi-level and decide man bedrooms and bathrooms for each floor. (User needs)

Data Model Design vs Implementation

- You need to build a home. So, how do we design this home?
  - Determine if the home is one level or multi-level and decide man bedrooms and bathrooms for each floor. (User needs)
  - Hire an architect to put the architecture in more detailed way for example, the size for each room, the distribution of the wireds, where the plumbing fixtures will be placed, etc. (Architecture phase)

Data Model Design vs Implementation

- You need to build a home. So, how do we design this home?
  - Determine if the home is one level or multi-level and decide man bedrooms and bathrooms for each floor. (User needs)
  - Hire an architect to put the architecture in more detailed way for example, the size for each room, the distribution of the wireds, where the plumbing fixtures will be placed, etc. (Architecture phase)
  - Decide the decorations, colors for each room, carpets, etc.

Data Model Design vs Implementation

- You need to build a home. So, how do we design this home?
  - Determine if the home is one level or multi-level and decide man bedrooms and bathrooms for each floor. (User needs)
  - Hire an architect to put the architecture in more detailed way for example, the size for each room, the distribution of the wireds, where the plumbing fixtures will be placed, etc. (Architecture phase)
  - Decide the decorations, colors for each room, carpets, etc.
- What do we do for the implementation?

Data Model Design vs Implementation

- You need to build a home. So, how do we design this home?
  - Determine if the home is one level or multi-level and decide man bedrooms and bathrooms for each floor. (User needs)
  - Hire an architect to put the architecture in more detailed way for example, the size for each room, the distribution of the wireds, where the plumbing fixtures will be placed, etc. (Architecture phase)
  - Decide the decorations, colors for each room, carpets, etc.
- What do we do for the implementation?
  - Hire a contractor to build (implement the design) the home.

Data Model Design vs Implementation

- You need to build a home. So, how do we design this home?
  - Determine if the home is one level or multi-level and decide man bedrooms and bathrooms for each floor. (User needs)
  - Hire an architect to put the architecture in more detailed way for example, the size for each room, the distribution of the wireds, where the plumbing fixtures will be placed, etc. (Architecture phase)
  - Decide the decorations, colors for each room, carpets, etc.
- What do we do for the implementation?
  - Hire a contractor to build (implement the design) the home.
  - This phase will implement the design but it also include some detail related to the actual way to build the tools and the material. (Physical Design)

DWH Characteristics

some details about hot vs cold storage,

Cold storage vs Hot storage

some details about hot vs cold storage,

Data Models

- Any Big Data solution working based distributed systems.

Data Models

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Introduction To Distributed Systems

Chapter Objectives

- Understand the distributed systems concepts.

Chapter Objectives

- Understand the distributed systems concepts.
- Replication and its usage in distributed systems.

Chapter Objectives

- Understand the distributed systems concepts.
- Replication and its usage in distributed systems.
- Partitioning and its usage in distributed systems .

Distributed Systems Concepts

- Any Big Data solution working based distributed systems.

Distributed Systems Concepts

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Distributed Systems Architecture

- Any Big Data solution working based distributed systems.

Distributed Systems Architecture

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Distributed Systems Challenges

- Any Big Data solution working based distributed systems.

Distributed Systems Challenges

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Design Simple Distributed System

- Any Big Data solution working based distributed systems.

Design Simple Distributed System

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Introduction to Hadoop and Map-Reduce

Chapter Objectives

- Introduction to Hadoop and its echo-systems.

Chapter Objectives

- Introduction to Hadoop and its echo-systems.
- Why we need Hadoop?

Chapter Objectives

- Introduction to Hadoop and its echo-systems.
- Why we need Hadoop?
- Understand the concept of HDFS and Map-Reduce.

Chapter Objectives

- Introduction to Hadoop and its echo-systems.
- Why we need Hadoop?
- Understand the concept of HDFS and Map-Reduce.
- Developing Map-Reduce applications.

Chapter Objectives

- Introduction to Hadoop and its echo-systems.
- Why we need Hadoop?
- Understand the concept of HDFS and Map-Reduce.
- Developing Map-Reduce applications.
- Using HiveQL over Map-Reduce.

Chapter Objectives

- Introduction to Hadoop and its echo-systems.
- Why we need Hadoop?
- Understand the concept of HDFS and Map-Reduce.
- Developing Map-Reduce applications.
- Using HiveQL over Map-Reduce.
- Hadoop advantages and disadvantages with use cases?

Hadoop Architecture

- Any Big Data solution working based distributed systems.

Hadoop Architecture

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Storage

- Any Big Data solution working based distributed systems.

Storage

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

YARN

- Any Big Data solution working based distributed systems.

YARN

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Hadoop I/O

- Any Big Data solution working based distributed systems.

Hadoop I/O

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Processing

- Any Big Data solution working based distributed systems.

Processing

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Map-Reduce

- Any Big Data solution working based distributed systems.

Map-Reduce

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Map-Reduce Components

- Any Big Data solution working based distributed systems.

Map-Reduce Components

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Word-Count Example

- Any Big Data solution working based distributed systems.

Word-Count Example

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Pig

- Any Big Data solution working based distributed systems.

Pig

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Hive

- Any Big Data solution working based distributed systems.

Hive

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

ZooKeeper

- Any Big Data solution working based distributed systems.

ZooKeeper

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Functional Programming

Spark Framework

Spark Framework: Spark Philosophy towards the Engine and the Programming languages

- Any Big Data solution working based distributed systems.

Spark Framework: Spark Philosophy towards the Engine and the Programming languages

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Spark Framework: Spark Basics

- Any Big Data solution working based distributed systems.

Spark Framework: Spark Basics

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Spark Basics

- Any Big Data solution working based distributed systems.

Spark Basics

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Spark Programming using RDDs

- Any Big Data solution working based distributed systems.

Spark Programming using RDDs

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Spark Programming using RDDs

- Any Big Data solution working based distributed systems.

Spark Programming using RDDs

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Spark Programming using RDDs

- Any Big Data solution working based distributed systems.

Spark Programming using RDDs

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Spark Datasets/Dataframe

- Any Big Data solution working based distributed systems.

Spark Datasets/Dataframe

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Spark Datasets/Dataframe

- Any Big Data solution working based distributed systems.

Spark Datasets/Dataframe

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Spark Datasets/Dataframe

- Any Big Data solution working based distributed systems.

Spark Datasets/Dataframe

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Spark on Production

- Any Big Data solution working based distributed systems.

Spark on Production

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Spark on Production

- Any Big Data solution working based distributed systems.

Spark on Production

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Spark For Batch Processing

- Any Big Data solution working based distributed systems.

Spark For Batch Processing

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Building custom input and output connector using Spark

- Any Big Data solution working based distributed systems.

Building custom input and output connector using Spark

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Spark Streaming

- Any Big Data solution working based distributed systems.

Spark Streaming

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Spark Streaming

- Any Big Data solution working based distributed systems.

Spark Streaming

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Spark Streaming

- Any Big Data solution working based distributed systems.

Spark Streaming

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Spark Streaming

- Any Big Data solution working based distributed systems.

Spark Streaming

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Spark using other Programming Languages

- Any Big Data solution working based distributed systems.

Spark using other Programming Languages

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Spark using other Programming Languages

- Any Big Data solution working based distributed systems.

Spark using other Programming Languages

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Spark using other Programming Languages

- Any Big Data solution working based distributed systems.

Spark using other Programming Languages

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Spark For Data Scientist

- Any Big Data solution working based distributed systems.

Spark For Data Scientist

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Spark For Data Scientist

- Any Big Data solution working based distributed systems.

Spark For Data Scientist

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Spark For Data Scientist

- Any Big Data solution working based distributed systems.

Spark For Data Scientist

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Spark Graph Dataframe/Graphx

- Any Big Data solution working based distributed systems.

Spark Graph Dataframe/Graphx

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Spark Graph Dataframe/Graphx

- Any Big Data solution working based distributed systems.

Spark Graph Dataframe/Graphx

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Tuning your Spark Jobs

- Any Big Data solution working based distributed systems.

Tuning your Spark Jobs

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Tuning your Spark Jobs

- Any Big Data solution working based distributed systems.

Tuning your Spark Jobs

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Real World Applications

Massaging Systems

Data Orchestration

NOSQL

Elastic

Data Architecture Design

Appendix

Appendix A- Shell Programming

- Any Big Data solution working based distributed systems.

Appendix A- Shell Programming

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Appendix B- Java Programming

- Any Big Data solution working based distributed systems.

Appendix B- Java Programming

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Appendix C- Scala Programming

- Any Big Data solution working based distributed systems.

Appendix C- Scala Programming

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Appendix D- SQL Programming

- Any Big Data solution working based distributed systems.

Appendix D- SQL Programming

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Appendix E- Oozie Orchestration

- Any Big Data solution working based distributed systems.

Appendix E- Oozie Orchestration

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Appendix F- DWH Concepts and Data Modeling Design

- Any Big Data solution working based distributed systems.

Appendix F- DWH Concepts and Data Modeling Design

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Appendix G- Machine Learning Concepts Data Engineers

- Any Big Data solution working based distributed systems.

Appendix G- Machine Learning Concepts Data Engineers

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?

Appendix H- Docker for Data Engineers

- Any Big Data solution working based distributed systems.

Appendix H- Docker for Data Engineers

- Any Big Data solution working based distributed systems.
- What is distributed systems in brief?