BIG DATA ANALYTICS

Unit 1

Unit1

Chapter 1: What Is Big Data and Why Is It Important? Challenges of Big Data.The Evolution of Data Management, Understanding the Waves of Managing Data, creating manageable data structures, Web and content management, Managing big data. DefiningBig Data, building a Successful Big Data Management Architecture, beginning with capture,organize, integrate, analyze, and act, Setting the architectural foundation, Performance matters, Traditional and advanced analytics.

Chapter 2: Examining Big Data Types and its Sources.

Defining Structured Data Exploring sources of big structured data, Understanding the role of relational databases in big data Defining Unstructured Data, exploring sources of unstructured data, Understanding the role of a CMS in big data management. Looking at Real-Time and Non-Real-Time Requirements, Putting Big Data Together, managing different data types, integrating data types into a big data environment.

Chapter 3: Technology Foundations of Big Data.

Exploring the Big Data Stack: - Layer 0: Redundant Physical Infrastructure – Physical redundant networks, Managing hardware: Storage and servers, Infrastructure operations -Layer 1: Security Infrastructure, Interfaces and Feeds to and from Applications and the Internet- Layer 2: Operational Databases. Layer 3: Organizing Data Services and Tools. Layer4: Analytical Data Warehouses, Big Data Analytics, Big Data Applications.

The Evolution of Data Management

- Although data management is typically viewed through a software lens, it actually has to be viewed from a holistic perspective.
- Data management has to include technology advances in hardware, storage, networking, and computing models such as virtualization and cloud computing.
- The convergence of emerging technologies and reduction in costs for everything from storage to compute cycles have transformed the data landscape and made new opportunities possible.
- Big data is the latest trend to emerge because of these factors

- Big data is defined as any kind of data source that has at least three shared characteristics:
- ✓ Extremely large Volumes of data
- ✓ Extremely high Velocity of data
- ✓ Extremely wide Variety of data
- Big data is important because it enables organizations to gather, store, manage, and manipulate vast amounts data at the right speed, at the right time, to gain the right insights.
- Big data is not a stand-alone technology; rather, it is a combination of the last 50 years of technology evolution.

Understanding the Waves of Managing Data

- Each data management wave is born out of the necessity to try and solve a specific type of data management problem.
- Each of these waves or phases evolved because of cause and effect. When a new technology solution came to market, it required the discovery of new approaches.
- When the relational database came to market, it needed a set of tools to allow managers to study the relationship between data elements.
- When companies started storing unstructured data, analysts needed new capabilities such as natural language–based analysis tools to gain insights that would be useful to business.

- The data management waves over the past five decades have culminated in where we are today: the initiation of the big data era.
- So, to understand big data, you have to understand the underpinning of these previous waves.
- You also need to understand that as we move from one wave to another
- we don’t throw away the tools and technology and practices that we have been using to address a different set of problems.

Wave 1: Creating manageable data structures

- As computing moved into the commercial market in the late 1960s, data was stored in flat files that imposed no structure.
- When companies needed to get to a level of detailed understanding about customers, they had to apply brute-force methods
- Later in the 1970s, things changed with the invention of the relational data model and the relational database management system (RDBMS)
- The relational model offered an ecosystem of tools from a large number of emerging software companies
- But a problem emerged from this exploding demand for answers: Storing this growing
- volume of data was expensive and accessing it was slow.

- When the volume of data that organizations needed to manage grew out of control, the data warehouse provided a solution.
- The data warehouse was intended to help companies deal with increasingly large amounts of structured data
- These data were analyzed by reducing the volume of the data to something smaller and more focused on a particular area of the business
- In addition, warehouses often store data from prior years for understanding organizational performance, identifying trends, and helping to expose patterns of behavior.
- Sometimes these data warehouses themselves were too complex and large and didn’t offer the speed and agility

- The answer was a further refinement of the data being managed through data marts.
- These data marts were focused on specific business issues
- Data Mart were much more streamlined and supported the business need for speedy queries than the more massive data warehouses
- Data warehouses and data marts solved many problems for companies needing a consistent way to manage massive transactional data
- But when it came to managing huge volumes of unstructured or semi-structured data, the warehouse was not able to evolve enough to meet changing demands.

- As companies began to store unstructured data, vendors began to add capabilities such as BLOBs (binary large objects).
- In essence, an unstructured data element would be stored in a relational database as one contiguous chunk of data.
- This object could be labeled (that is, a customer inquiry) but you couldn’t see what was inside that object.
- The object database stored the BLOB as an addressable set of pieces so that we could see what was in there

- Unlike the BLOB, the object database provided a unified approach for dealing with unstructured data
- Object databases include a programming language and a structure for the data elements
- So it was easier to manipulate various data objects without programming and complex joins.
- The object databases introduced a new level of innovation that helped lead to the second wave of data management.

Wave 2: Web and content management

- In the 1990s with the rise of the web, organizations wanted to move beyond documents and store and manage web content, images, audio, and video
- The market evolved from a set of disconnected solutions to a more unified model
- This brought business process management, version control, information recognition, text management, and collaboration elements together
- This new generation of systems added metadata (information about the organization and characteristics of the stored information).

- These solutions remain incredibly important for companies needing to manage all this data in a logical manner.
- But at the same time, a new generation of requirements has begun to emerge that drive us to the next wave.
- These new requirements have been driven, in large part, by a convergence of factors including the web, virtualization, and cloud computing.
- In this new wave, organizations are beginning to understand that they need to manage a new generation of data sources in an unprecedented amount
- A variety of data that needs to be processed at an unheard-of speed.

Wave 3: Managing big data

- With big data, it is now possible to virtualize data so that it can be stored efficiently
- In addition, improvements in network speed and reliability have removed other physical limitations of being able to manage massive amounts of data at an acceptable pace.
- Add to this the impact of changes in the price and sophistication of computer memory.
- Many of the technologies at the heart of big data, such as virtualization, parallel processing, distributed file systems, and in-memory databases, have been around for decades

- Other technologies such as Hadoop and MapReduce have been on the scene for only a few years.
- This combination of technology advances can now address significant business problems.
- Businesses want to be able to gain insights and actionable results from many different kinds of data at the right speed — no matter how much data is involved.
- The move to big data is not just about businesses. Science, research, and government activities have also helped to drive it forward.

- Different approaches to handling data exist based on whether it is data in motion or data at rest.
- Data in motion would be used if a company is able to analyze the quality of its products during the manufacturing process to avoid costly errors.
- Data at rest would be used by a business analyst to better understand customers' current buying patterns
- These patters are based on all aspects of the customer relationship, including sales, social media data, and customer service interactions.

Defining Big Data

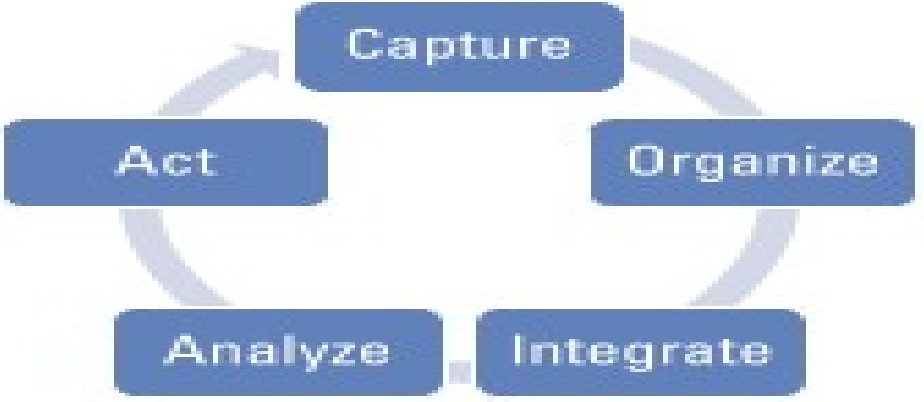
- Big data is not a single technology but a combination of old and new technologies that helps companies gain actionable insight.
- Therefore, big data is he capability to manage a huge volume of disparate data, at the right speed, and within the right time frame to allow real-time analysis and reaction.
- big data is typically broken down by three characteristics:
- ✓Volume: How much data
- ✓ Velocity: How fast that data is processed
- ✓ Variety: The various types of data

Building a Successful Big Data Management Architecture

- It is necessary to identify the right amount and types of data that can be analyzed to impact business outcomes
- Big data incorporates all data, including structured data and unstructured data from e-mail,social media, text streams, and more.
- This kind of data management requires that companies leverage both their structured and unstructured data.
- But as data has become the fuel of growth and innovation, it is more important than ever to have an underlying architecture to support growing requirements.

Beginning with capture, organize, integrate, analyze, and act

- Before we delve into the architecture, it is important to take into account the functional requirements for big data.
- Figure below illustrates that data must first be captured, and then organized and integrated.



- After this phase is successfully implemented, data can be analyzed based on the problem being addressed.
- Finally, management takes action based on the outcome of that analysis.
- For example, Amazon.com might recommend a book based on a past purchase or a customer might receive a coupon for a discount for a future purchase of a related product to one that was just purchased.
- If your organization is combining data sources, it is critical that you have the ability to validate that these sources make sense when combined

Setting the architectural foundation

- In addition to supporting the functional requirements, it is important to support the required performance.
- Your needs will depend on the nature of the analysis you are supporting. You will need the right amount of computational power and speed.
- While some of the analysis you will do will be performed in real time, you will inevitably be storing some amount of data as well.
- Your architecture also has to have the right amount of redundancy so that you are protected from unanticipated latency and downtime.

- Your organization and its needs will determine how much attention you have to pay to these performance issues.
- So, start out by asking yourself the following questions:
- ✓ How much data will my organization need to manage today and in the future?
- ✓ How often will my organization need to manage data in real time or near
- real time?
- ✓ How much risk can my organization afford?
- ✓ How important is speed to my need to manage data?
- ✓ How certain or precise does the data need to be?

- To understand big data, it helps to lay out the components of the architecture.
- A big data management architecture must include a variety of services that enable companies to make use of myriad data sources in a fast and effective manner.
- To help you make sense of this, we put the components into a diagram that will help you see what’s there and the relationship between the components.
- In the next section, we explain each component and describe how these components are related to each other.

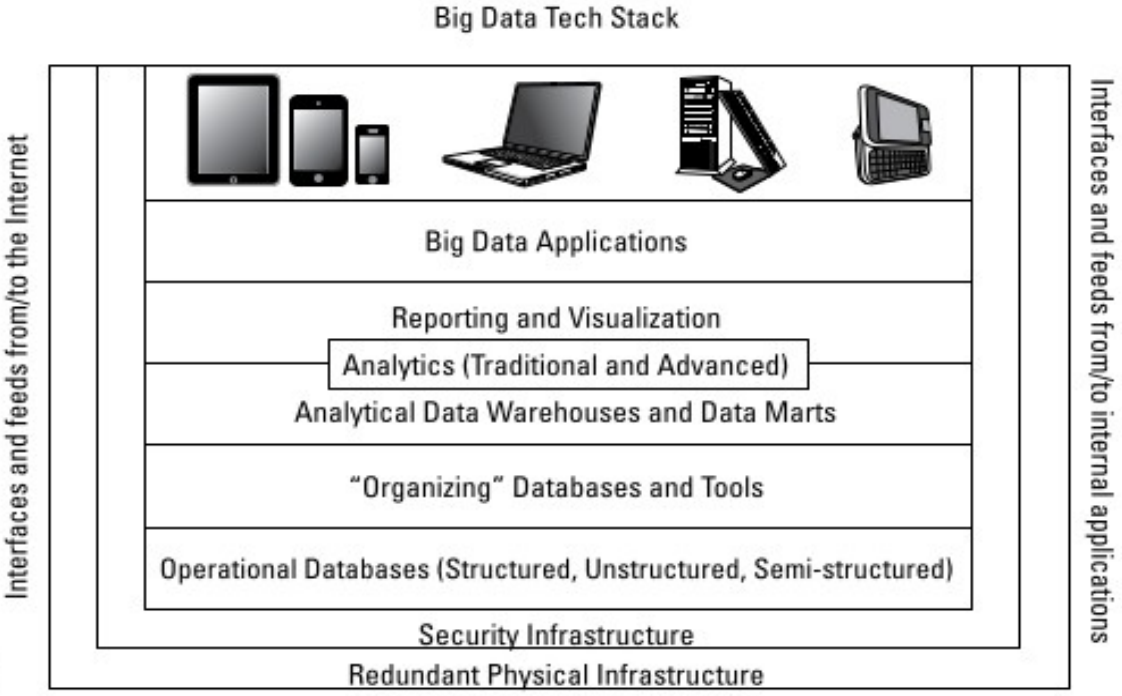


Figure 1-2: The big data architecture.

- Although it’s convenient to simplify big data into the three Vs, it can be misleading and overly simplistic.
- For example, you may be managing a relatively small amount of very disparate, complex data or you may be processing a huge volume of very simple data.
- That simple data may be all structured or all unstructured. Even more important is the fourth V: veracity.
- How accurate is that data in predicting business value? Do the results of a big data analysis actually make sense

Interfaces and feeds

- To understand how big data works in the real world, it is important to start by understanding this necessity.
- In fact, what makes big data big is the fact that it relies on picking up lots of data from lots of sources.
- Therefore, open application programming interfaces (APIs) will be core to any big data architecture.
- In addition, keep in mind that interfaces exist at every level and between every layer of the stack. Without integration services, big data can’t happen.

Redundant physical infrastructure

- Redundancy is important because we are dealing with so much data from so many different sources. Redundancy comes in many forms.
- If your company has created a private cloud, you will want to have redundancy built within the private environment so that it can scale out to support changing Workloads.
- If your company wants to contain internal IT growth, it may use external cloud services to augment its internal resources.
- In some cases, this redundancy may come in the form of a Software as a Service (SaaS) offering that allows companies to do sophisticated data analysis as a service.
- The SaaS approach offers lower costs, quicker startup, and seamless evolution of the underlying technology.

Security infrastructure

- The more important big data analysis becomes to companies, the more important it will be to secure that data.
- For example, if you are a healthcare company, you will probably want to use big data applications to determine changes in demographics or shifts in patient needs.
- You will need to take into account who is allowed to see the data and under what circumstances they are allowed to do so.
- You will need to be able to verify the identity of users as well as protect the identity of patients.
- These types of security requirements need to be part of the big data fabric from the outset and not an afterthought.

Operational data sources

- Traditionally, an operational data source consisted of highly structured data managed by the line of business in a relational database.
- But as the world changes, it is important to understand that operational data now has to encompass a broader set of data sources
- These includes unstructured sources such as customer and social media data in all its forms
- You find new emerging approaches to data management in the big data world, including document, graph, columnar, and geospatial database architectures.
- You need to include both relational databases and nonrelational databases in your approach to harnessing big data.

- All these operational data sources have several characteristics in common:
- ✓ They represent systems of record that keep track of the critical data required for real-time, day-to-day operation of the business.
- ✓ They are continually updated based on transactions happening within business units and from the web.
- ✓ For these sources to provide an accurate representation of the business, they must blend structured and unstructured data.
- ✓ These systems also must be able to scale to support thousands of users on a consistent basis.
- ✓These might include transactional e-commerce systems, customer relationship management systems, or call center applications.

Performance matters

- The data architecture needs to perform in concert with your organization’s supporting infrastructure.
- For example, you might be interested in running models to determine whether it is safe to drill for oil in an offshore area given real-time data of temperature
- It might take days to run this model using a traditional server configuration.
- However, using a distributed computing model, what took days might now take minutes.

- Performance determine the kind of database you would use.
- For example, in some situations, you may want to understand how two very distinct data elements are related.
- What is the relationship between buzz on a social network and the growth in sales? This is not the typical query you could ask of a structured, relational database.
- A graphing database might be a better choice, as it is specifically designed to separate the “nodes” or entities from its “properties”
- Typically the graph database will be used in scientific and technical applications.

- Other important operational database approaches include columnar data- bases that store information efficiently in columns rather than rows.
- This approach leads to faster performance because input/output is extremely fast.
- When geographic data storage is part of the equation, a spatial database is optimized to store and query data based on how objects are related in space.

Organizing data services and tools

- A growing amount of data comes from a variety of sources that aren’t quite as organized or straightforward, including data that comes from machines or sensors, and massive public and private data sources.
- In the past, most companies weren’t able to either capture or store this vast amount of data. It was simply too expensive or too overwhelming.
- Even if companies were able to capture the data, they did not have the tools to do anything about it. Very few tools could make sense of these vast amounts of data.
- The tools that did exist were complex to use and did not produce results in a reasonable time frame.
- This has the undesir able effect of missing important events because they were not in a particular snapshot.

MapReduce, Hadoop, and Big Table

- With the evolution of computing technology, it is now possible to manage immense volumes of data that previously could have only been handled by supercomputers at great expense.
- Prices of systems have dropped, and as a result, new techniques for distributed computing are mainstream.
- The real breakthrough in big data happened as companies like Yahoo!, Google, and Facebook monetizing the massive amounts of data their offerings were creating.
- In particular, the innovations MapReduce, Hadoop, and Big Table proved to be the sparks that led to a new generation of data management.
- These technologies address one of the most fundamental problems — the capability to process massive amounts of data efficiently, costeffectively, and in a timely fashion.

MapReduce

- MapReduce was designed by Google as a way of efficiently executing a set of functions against a large amount of data in batch mode.
- The “map” component distributes the programming problem or tasks across a large number of systems
- It handles the placement of the tasks in a way that balances the load and manages recovery from failures.
- After the distributed computation is completed, another function called “reduce” aggregates all the elements back together to provide a result.
- An example of MapReduce usage would be to determine how many pages of a book are written in each of 50 different languages.

Big Table

- Big Table was developed by Google to be a distributed storage system intended to manage highly scalable structured data.
- Data is organized into tables with rows and columns.
- Unlike a traditional relational database model, Big Table is a sparse, distributed, persistent multidimensional sorted map.
- It is intended to store huge volumes of data across commodity servers.

Hadoop

- Hadoop is an Apache-managed software framework derived from MapReduce
- and Big Table.
- Hadoop allows applications based on MapReduce to run on large clusters of commodity hardware.
- The project is the foundation for the computing architecture supporting Yahoo!’s business.
- Hadoop is designed to parallelize data processing across computing nodes to speed computations and hide latency.
- Two major components of Hadoop exist: a massively scalable distributed file system that can support petabytes of data and a massively scalable MapReduce engine that computes results in batch

Traditional and advanced analytics

- It requires many different approaches to analysis, depending on the problem being solved.
- Some analyses will use a traditional data warehouse, while other analyses will take advantage of advanced predictive analytics.
- Managing big data holistically requires many different approaches to help the business to successfully plan for the future.

- **Analytical data warehouses and data marts:**

- After a company sorts through the massive amounts of data available, it is often pragmatic to take the subset of data that reveals patterns and put it
- into a form that’s available to the business.
- These warehouses and marts provide compression, multilevel partitioning, and a massively parallel processing architecture.

- **Big data analytics:**

- The capability to manage and analyze petabytes of data enables companies to deal with clusters of information that could have an impact on the business.
- This requires analytical engines that can manage this highly distributed data and provide results that can be optimized to solve a business problem.
- Analytics can get quite complex with big data
- For example, some organizations are using predictive models that couple structured and unstructured data together to predict fraud.

- **Reporting and visualization**
- Organizations have always relied on the capability to create reports to give them an understanding on projections of growth.
- Big data changes the way that data is managed and used.
- If a company can collect, manage, and analyze enough data, it can use a new generation of tools to analyze them
- This help management truly understand the context based on the business problem being addressed.
- With big data, reporting and data visualization become tools for looking at the context of how data is related and the impact of those relationships on the future.

- **Big data applications:**
- Some of the emerging applications are in areas such as healthcare, manufacturing management, traffic management, and so on
- They rely on huge volumes, velocities, and varieties of data to transform the behavior of a market
- In healthcare, a big data application might be able to monitor premature infants to determine when data indicates when intervention is needed
- In manufacturing, a big data application can be used to prevent a machine from shutting down during a production run
- A big data traffic management application can reduce the number of traffic jams on busy city highways to decrease accidents, save fuel, and reduce pollution.

Defining Structured Data

- The term structured data generally refers to data that has a defined length and format.
- Examples of structured data include numbers, dates, and groups of words and numbers called strings (for example, a customer’s name, address, and so on).
- Most experts agree that this kind of data accounts for about 20 percent of the data that is out there.
- Structured data is the data that you’re probably used to dealing with. It’s usually stored in a database.
- Example might include your customer relationship management (CRM) data, operational enterprise resource planning (ERP) data, and financial data.
- Often these data elements are integrated in a data warehouse for analysis.

Exploring sources of big structured data

- Although this might seem like business as usual, in reality, structured data is taking on a new role in the world of big data.
- The evolution of technology provides newer sources of structured data being produced — often in real time and in large volumes.
- The sources of data are divided into two categories:
- ✓ Computer- or machine-generated: Machine-generated data generally refers to data that is created by a machine without human intervention.
- ✓ Human-generated: This is data that humans, in interaction with computers, supply.
- Some experts argue that a third category exists that is a hybrid between machine and human. Here though, we’re concerned with the first two categories.

- Machine-generated structured data can include the following:
- ✓ **Sensor data**: Examples include radio frequency ID (RFID) tags, smart meters, medical devices, and Global Positioning System (GPS) data.
- For example, RFID is rapidly becoming a popular technology. It uses tiny computer chips to track items at a distance.
- ✓ **Web log data**: When servers, applications, networks, and so on operate, they capture all kinds of data about their activity.This can amount to huge volumes of data that can be useful
- For example, to deal with service-level agreements or to predict security breaches.
- ✓ **Point-of-sale data**: When the cashier swipes the bar code of any product that you are purchasing, all that data associated with the product is generated.
- ✓ **Financial data:** Lots of financial systems are now programmatic; they are operated based on predefined rules that automate processes. Stock- trading data is a good example of this

- Examples of structured human-generated data might include the following:
- ✓ **Input data:** This is any piece of data that a human might input into a computer, such as name, age, income, non-free-form survey responses,and so on. This data can be useful to understand basic customer behavior.
- ✓ **Click-stream data:** Data is generated every time you click a link on a website. This data can be analyzed to determine customer behavior and buying patterns.
- ✓ **Gaming-related data:** Every move you make in a game can be recorded. This can be useful in understanding how end users move through a gaming portfolio.

Understanding the role of relational databases in big data

- Data persistence refers to how a database retains versions of itself when Modified.
- The great granddaddy of persistent data stores is the relational database management system (RDBMS).
- In its infancy, the computing industry used what are now considered primitive techniques for data persistence.
- Understanding the relational database is important because other types of databases are used with big data.
- In a relational model, the data is stored in a table. This database would contain a schema — that is, a structural representation of what is in the database.

- For example, in a relational database, the schema defines the tables,the fields in the tables, and the relationships between the two.
- The data is stored in columns, one each for each specific attribute, also stored in the rows.
- For instance, the two tables shown in Figure 2-1 represent the schema for a simple database.
- The first table stores product information; the second stores demographic information. Each has various attributes
- Each table can be updated with new data, and data can be deleted, read, and updated.
- This is often accomplished in a relational model using a structured query language (SQL).

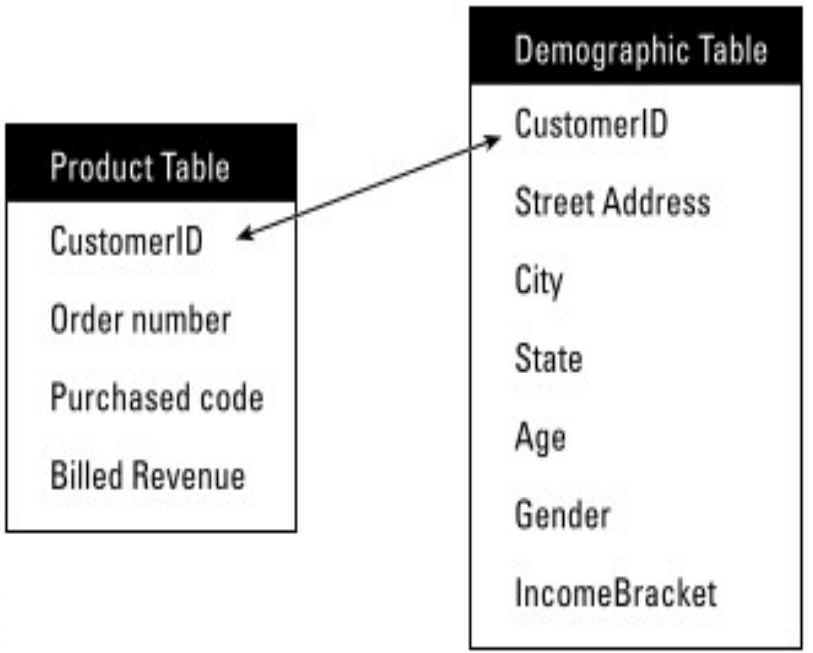


Figure 2-1: The relationships between tables.

- Another aspect of the relational model using SQL is that tables can be queried using a common key (that is, the relationship).
- In Figure 2-1, the common key in the tables is CustomerID.
- You can submit a query, for example, to determine the gender of customers who purchased a specific product. It might look something like this:
- **Select CustomerID, State, Gender, Product from “demographic table”, “product table” where Product= XXYY**
- Although relational databases have ruled the roost for the last several decades, they can be difficult to use when you’re dealing with huge streams of disparate data types.

Defining Unstructured Data

- Unstructured data is data that does not follow a specified format.
- If 20 percent of the data available to enterprises is structured data, the other 80 percent is unstructured.
- Unstructured data is really most of the data that you will encounter.
- Until recently, however, the technology didn’t really support doing much with it except storing it or analyzing it manually.

Exploring sources of unstructured data

- Unstructured data is everywhere. In fact, most individuals and organizations conduct their lives around unstructured data.
- Just as with structured data, unstructured data is either machine generated or human generated.
- Here are some examples of machine-generated unstructured data:
- ✓ **Satellite images**: This includes weather data or the data that the government captures in its satellite surveillance imagery. Just think about Google Earth, and you get the picture (pun intended).
- ✓ **Scientific data:** This includes seismic imagery, atmospheric data, and high energy physics.
- ✓ **Photographs and video:** This includes security, surveillance, and traffi video.
- ✓ **Radar or sonar data:** This includes vehicular, meteorological, and oceanographic seismic profiles.

- The following list shows a few examples of human-generated unstructured data:
- ✓ **Text internal to your company:** Think of all the text within documents, logs, survey results, and e-mails. Enterprise information actually repre-sents a large percent of the text information in the world today.
- ✓ **Social media data:** This data is generated from the social media platforms such as YouTube, Facebook, Twitter, LinkedIn, and Flickr.
- ✓ **Mobile data:** This includes data such as text messages and location information.
- ✓ **Website content:** This comes from any site delivering unstructured content, like YouTube, Flickr, or Instagram.

Understanding the role of a CMS in big data management

- Organizations store some unstructured data in databases.
- However, they also utilize enterprise content management systems (CMSs) that can manage the complete life cycle of content.
- This can include web content, document content, and other forms media.
- Systems that are designed to store content in the form of content management systems are no longer stand-alone solutions.
- Rather, they are likely to be part of an overall data management solution.

- For example, your organization may monitor Twitter feeds that can then programmatically trigger a CMS search
- Now, the person who triggered the tweet gets an answer back that offers a location where the individual can find the product that he or she might be looking for.
- The greatest benefit is when this type of interaction can happen in real time.
- It also illustrates the value of leveraging real-time unstructured, structured (customer data about the person who tweeted), and semi-structured (the actual content in the CMS) data

- The following list highlights a few things you need to consider regarding a system’s capability to ingest data,process it, and analyze it in real time.
- ✓ **Low latency:**
- Latency is the amount of time lag that enables a service to execute in an environment.
- Some applications require less latency,which means that they need to respond in real time.
- A real-time stream is going to require low latency.
- So you need to be thinking about compute power as well as network constraints.

- ✓ **Scalability**: Scalability is the capability to sustain a certain level of performance even under increasing loads.
- ✓ **Versatility**: The system must support both structured and unstructured data streams.
- ✓ **Native format:** Use the data in its native form. Transformation takes time and money. The capability to use the idea of processing complex interactions in the data that trigger events may be transformational.

Looking at Real-Time and Non-Real-Time Requirements

- The big change that with big data is the capability to leverage massive amounts of data without all the complex programming that was required in the past.
- Big data approaches will help keep things in balance so we don’t go over the edge as the volume, variety, and velocity of data changes.
- The real-time aspects of big data can be revolutionary when companies need to solve significant problems.
- In general, this real-time approach is most relevant when the answer to a problem is time sensitive and business critical.

- The following list shows examples of when a company wants to leverage this real-time data to gain a quick advantage:
- ✓ Monitoring for an exception with a new piece of information, like fraud/intelligence
- ✓ Monitoring news feeds and social media to determine events that may impact financial markets, such as a customer reaction to a new product announcement
- ✓ Changing your ad placement during a big sporting event based on real-time Twitter streams
- ✓ Providing a coupon to a customer based on what he bought at the point of sale

Putting Big Data Together

- What you want to do with your structured and unstructured data indicates
- why you might choose one piece of technology over another one.
- It also determines the need to understand inbound data structures to put this data in the right place.

Managing different data types

- Figure 2-2 shows a helpful table that outlines some of the characteristics of big data and the types of data management systems you might want to use to address each one.

	Batch	Streaming	Complex Query
Structured	Hadoop	Key/Value	RDBMS
Unstructured	Document	Graph Spatial	Columnar
Both	Hybrid	Hybrid	Hybrid

Figure 2-2: The characteristics of different data types.

Integrating data types into a big data environment

- Oftentimes, it becomes necessary to integrate different sources.
- This data may be coming from all internal systems, from both internal and external sources, or from entirely external sources.
- Components needed to include connectors and metadata, to catch data in real time.

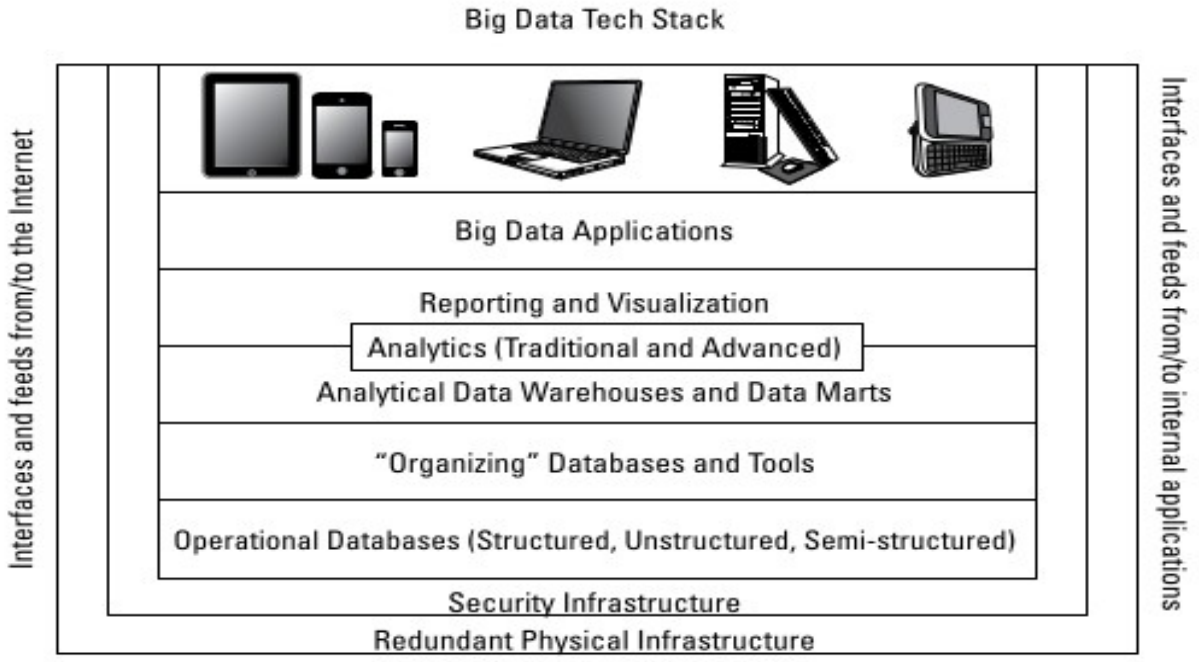
- **Connectors:**
- connectors enables to pull data in from various big data sources.
- Maybe you want a Twitter connector or a Facebook one.
- Maybe you need to integrate from your data warehouse with a big data source that’s off your premises so that you can analyze both of these sources of data together.

- **Metadata**
- A critical component to integrating all this data is the metadata.
- Metadata is the definitions, mappings, and other characteristics used to describe how to find, access, and use a company’s data (and software) components.
- One example of metadata is data about an account number.
- This might include the number, description, data type, name, address, phone number, and privacy level.

- Metadata can be used to help you organize your data stores and deal with new and changing sources of data.
- Although the idea of metadata is not new, it is changing and evolving in the context of big data.
- In the traditional meta data world, it is important to have a catalog that provides a single view of all data sources.
- You may need an analytic tool that will help you understand the underlying metadata.

Technology Foundations for Big Data

Exploring the Big Data Stack

- 

Layer 0: Redundant Physical Infrastructure

- At the lowest level of the stack is the physical infrastructure — the hardware, network, and so on.
- Big data implementations have very specific requirements on all elements in the reference architecture
- so we need to examine these requirements on a layer-by-layer basis to ensure that implementation will perform and scale according to the demands of business.
- A prioritized list of these principles should include statements about the following:

- ✓ **Performance**:
- How responsive do you need the system to be?
- Performance, also called latency, is often measured end to end, based on a single transaction or query request.
- Very fast (high-performance, low-latency) infrastructures tend to be very expensive.

- ✓ **Availability**:
- Do you need a 100 percent uptime guarantee of service?
- How long can your business wait in the case of a service interruption or failure?
- Highly available infrastructures are also very expensive.

- ✓ **Scalability**:
- How big does your infrastructure need to be?
- How much disk space is needed today and in the future?
- How much computing power do you need?

- ✓ **Flexibility**:
- How quickly can you add more resources to the infrastructure Ture?
- How quickly can your infrastructure recover from failures?
- The most flexible infrastructures can be costly, but you can control the costs
- with cloud services, where you only pay for what you actually use

- ✓ **Cost**: What can you afford?
- Because the infrastructure is a set of components, you might be able to buy the “best” networking and decide to save money on storage (or vice versa).
- You need to establish requirements for each of these areas in the context of an overall budget and then make trade-offs where necessary
- As big data is all about high-velocity, high-volume, and high-data variety, the physical infrastructure will literally “make or break” the implementation.
- Most big data implementations need to be highly available, so the networks, servers, and physical storage must be both resilient and redundant.

- Resiliency and redundancy are interrelated.
- An infrastructure, or a system,is resilient to failure or changes when sufficient redundant resources are in place, ready to jump into action.
- In essence, there are always reasons why even the most sophisticated and resilient network could fail, such as a hardware malfunction.
- Resiliency helps to eliminate single points of failure in your infrastructure.

- **For example,** if only one network connection exists between your busines and the Internet, no network redundancy exists, and the infrastructure is not resilient with respect to a network outage

Physical redundant networks

- Networks should be redundant and must have enough capacity to accommodate the anticipated volume and velocity of the inbound and outbound data
- As you begin making big data an integral part of your computing strategy, it is reasonable to expect volume and velocity to increase.
- Infrastructure designers should plan for these expected increases and try to create physical implementations that are “elastic.”
- As network traffic ebbs and flows, so too does the set of physical assets associated with the implementation.
- Your infrastructure should offer monitoring capabilities so that operators can react when more resources are required to address changes in workloads

Managing hardware: Storage and servers

- Likewise, the hardware (storage and server) assets must have sufficient speed and capacity to handle all expected big data capabilities.
- It’s of little use to have a high-speed network with slow servers because the servers will most likely become a bottleneck.
- However, a very fast set of storage and compute servers can overcome variable network performance.
- Of course, nothing will work properly if network performance is poor or unreliable.

Infrastructure operations

- Another important design consideration is infrastructure operations management.
- The greatest levels of performance and flexibility will be present only in a well-managed environment.
- Data center managers need to be able to anticipate and prevent catastrophic failures so that the integrity of the data,and by extension the business processes, is maintained.
- IT organizations often overlook and therefore underinvest in this area.

Layer 1: Security Infrastructure

- Security and privacy requirements for big data are similar to the requirements for conventional data environments.
- Some unique challenges arise when big data becomes part of the strategy, which we briefly describe in this list:
- ✓ **Data access:**
- User access to raw or computed big data has about the same level of technical requirements as non-big data implementations.
- The data should be available only to those who have a legitimate business need for examining or interacting with it.
- Most core data storage platforms have rigorous security schemes and are often augmented with a federated identity capability, providing appropriate access across the many layers of the architecture.

- ✓ **Application access:**
- Application access to data is also relatively straightforward from a technical perspective.
- Most application programming interfaces (APIs) offer protection from unauthorized usage or access.
- This level of protection is probably adequate for most big data implementations.
- ✓ **Threat detection:**
- The inclusion of mobile devices and social networks
- exponentially increases both the amount of data and the opportunities
- for security threats. It is therefore important that organizations take a
- multiperimeter approach to security.

- ✓ **Data encryption:**
- Data encryption is the most challenging aspect of security in a big data environment.
- In traditional environments, encrypting and decrypting data really stresses the systems' resources.
- With the volume, velocity, and varieties associated with big data, this problem is exacerbated.
- The simplest (brute-force) approach is to provide more and faster computational capability.
- However, this comes with a steep price tag — especially when you have to accommodate resiliency requirements.
- A more temperate approach is to identify the data elements requiring this level of security and to encrypt only the necessary items.

Interfaces and Feeds to and from Applications and the Internet

- The next level in the stack is the interfaces that provide bidirectional access to all the components of the stack — from corporate applications to data feeds from the Internet.
- For decades, programmers have used APIs to provide access to and from software implementations
- API toolkits have a couple of advantages over internally developed APIs.
- Big data challenges require a slightly different approach to API development or adoption.
- Because much of the data is unstructured and is generated outside of the control of your business, a new technique, called Natural Language Processing (NLP)

- NLP allows you to formulate queries with natural language syntax instead of a formal query language like SQL.
- One way to deal with interfaces is to implement a “connector” factory.
- This connector factory adds a layer of abstraction and predictability to the process, and it leverages many of the lessons and techniques used in Service Oriented Architecture (SOA).
- To create as much flexibility as necessary, the factory could be driven with interface
- descriptions written in Extensible Markup Language (XML).
- This level of abstraction allows specific interfaces to be created easily and quickly with-
- out the need to build specific services for each data source.

Layer 2: Operational Databases

- At the core of any big data environment are the database engines containing the collections of data elements relevant to your business.
- These engines need to be fast, scalable, and rock solid.
- They are not all created equal, and certain big data environments will fare better with one engine than another,or more likely with a mix of database engines.
- For example, although it is possible to use relational database management systems (RDBMSs) for all your big data implementations, it is not practical to do so because of performance, scale, or even cost.
- SQL is the most prevalent database query language in use today, than other languages

- It is very important to understand what types of data can be manipulated by the database and whether it supports true transactional behavior.
- Database designers describe this behavior with the acronym ACID. It stands for
- ✓ **Atomicity**: A transaction is “all or nothing” when it is atomic. If any part of the transaction or the underlying system fails, the entire transaction fails.
- ✓ **Consistency**: Only transactions with valid data will be performed on the database.
- ✓ **Isolation**: Multiple, simultaneous transactions will not interfere with each other.
- ✓ **Durability**: After the data from the transaction is written to the database, it stays there “forever.”

- Table 4-1 offers a comparison of these characteristics of SQL and NoSQL databases.

Table 4-1 offers a comparison of these characteristics of SQL and NoSQL databases.

Table 4-1 Important Characteristics of SQL and NoSQL Databases

Engine	*Query Language*	*MapReduce*	*Data Types*	*Transactions*	*Examples*
Relational	SQL, Python, C	No	Typed	ACID	PostgreSQL, Oracle, DB/2
Columnar	Ruby	Hadoop	Predefined and typed	Yes, if enabled	HBase
Graph	Walking, Search, Cypher	No	Untyped	ACID	Neo4J
Document	Commands	JavaScript	Typed	No	MongoDB, CouchDB
Key-value	Lucene, Commands	JavaScript	BLOB, semityped	No	Riak, Redis

Layer 3: Organizing Data Services and Tools

- Organizing data services and tools capture, validate, and assemble various big data elements into contextually relevant collections.
- Because big data is massive, techniques have evolved to process the data efficiently and seamlessly.
- Organizing data services are, in reality, an ecosystem of tools and technologies that can be used to gather and assemble data in preparation for further processing.
- As such, the tools need to provide integration, translation, normalization, and scale. Technologies in this layer include the following:

- ✓ **A distributed file system:** Necessary to accommodate the decomposition of data streams and to provide scale and storage capacity
- ✓ **Serialization services**: Necessary for persistent data storage and multilanguage remote procedure calls (RPCs)
- ✓ **Coordination services:** Necessary for building distributed applications(locking and so on)
- ✓ **Extract, transform, and load (ETL) tools:** Necessary for the loading and conversion of structured and unstructured data into Hadoop
- ✓ **Workflow services:** Necessary for scheduling jobs and providing a structure for synchronizing process elements across layers

Layer 4: Analytical Data Warehouses

- The data warehouse , and its companion the data mart, are used to optimize data to help decision makers.
- Typically, data warehouses and marts contain normalized data gathered from a variety of sources
- Data warehouses and marts simplify the creation of reports and the visualization of disparate data items.
- They are generally created from relational databases, multidimensional databases, flat files, and object database

- Most data warehouse implementations are kept current via batch processing
- Batch-loaded data warehouses and data marts may be insufficient for many big data applications.
- The stress imposed by high-velocity data streams will likely require a more real-time approach to big data warehouses.
- The performance and scale will reflect the time requirements of the analysts and decision makers.

- Data warehouses and data marts are comprised of data gathered from various sources the costs associated with the cleansing and normalizing of the data must also be addressed.
- with big data, you find some key differences:
- ✓ Traditional data streams (from transactions, applications, and so on) can produce a lot of disparate data.
- ✓ Dozens of new data sources also exist, each of them needing some degree of manipulation before it can be timely and useful to the business.
- ✓ Content sources will also need to be cleansed, and these may require different techniques than you might use with structured data.

Big Data Analytics

- Existing analytics tools and techniques will be very helpful in making sense of big data. However, there is a catch.
- The algorithms that are part of these tools have to be able to work with large amounts of potentially real-time and disparate data.
- vendors providing analytics tools will also need to ensure that their algorithms work across distributed implementations.
- Three classes of tools are listed in this layer of reference architecture.
- They can be used independently or collectively by decision makers to help steer the business.

- The three classes of tools are as follows:
- ✓ **Reporting and dashboards:**
- These tools provide a “user-friendly” representation of the information from various sources.
- Although a mainstay in the traditional data world, this area is still evolving for big data.
- Some of the tools that are being used are traditional ones that can now access the new kinds of databases collectively called NoSQL (Not Only SQL)

- ✓ **Visualization:**
- These tools are the next step in the evolution of reporting.
- The output tends to be highly interactive and dynamic in nature.
- Another important distinction between reports and visualized output is animation.
- Business users can watch the changes in the data utilizing a variety of different visualization techniques
- These includes mind maps,heat maps, infographics, and connection diagrams.

- ✓ **Analytics and advanced analytics:**
- These tools reach into the data warehouse and process the data for human consumption.
- Advanced analytics should explicate trends or events that are transformative, unique,or revolutionary to existing business practice.
- Predictive analytics and sentiment analytics are good examples of this science.

Big Data Applications

- Custom and third-party applications offer an alternative method of sharing and examining big data sources.
- Although all the layers of the reference architecture are important this layer is where most of the innovation and creativity is evident.
- These applications are either horizontal, in that they address problems that are common across industries
- The most prevalent categories as of this writing are log data applications (Splunk, Loggly), ad/media applications (Bluefin, DataXu),and marketing applications (Bloomreach, Myrrix).

- The creation of big data applications will require structure, standards, rigor, and well-defined APIs.
- Most business applications wanting to leverage big data will need to
- subscribe to APIs across the entire stack.
- It may be necessary to process raw data from the low-level data stores and combine the raw data with synthesized output from the warehouses.
- software development teams need to be able to rapidly create applications germane to solving the business challenge of the moment.

- Companies may need to think about creating development teams which rapidly respond to changes in the business environment
- These development teams must create and deploy applications on demand.
- These applications are considered as “semicustom” because they involve more assembly than actual low-level coding.
- Software developers need to standardized development environments and devise for rapid rollout of big data applications