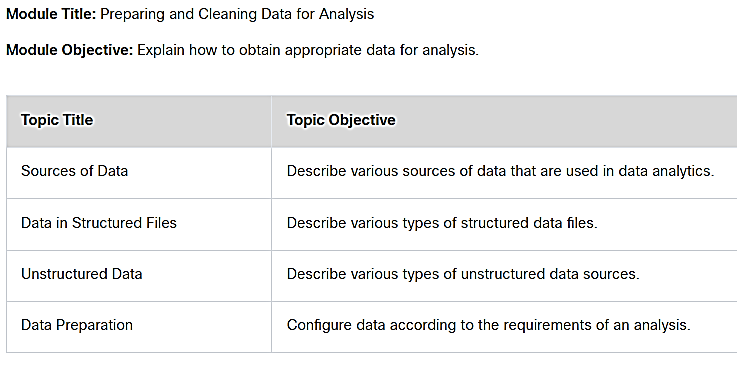
**Module-III**

**Preparing and Cleaning Data for Analysis**

****

**3.1 Sources of Data**

**3.1.2 Selecting Relevant Data**

Selecting relevant data for your analysis **includes determining the type(s)** of data that you need and **finding a source** for the data. When selecting data for a project, it is important to focus on **finding data that may provide insights into your original business question**. *For example, if you are seeking to understand demographic characteristics of people who bought Product X in the past year, you should only be using data that is directly related to Product X.* This process is crucial to ensuring the validity and reliability of your analysis. **Sometimes** the **data** you need to answer your questions **isn’t readily available**. It may be necessary to establish new procedures to **collect the data** required for your analysis. Other times, it may involve **combining data** from multiple sources into a format that can be analyzed.

**Let’s take** the example of an entertainment producer gathering data about the **viability of a movie project**. **If** the movie is an **adaptation of a book**, they need **data on the sales of books** by that author, *within that genre, and across a variety of population demographics*. They might compare the profitability of other movies with similar plots or characters, and their release dates, to determine the best time of the year to release a picture of that genre. Producers **may also** analyze data on **the actors and locations** that appear in the most successful recent movies to make casting and production decisions.

Some questions that you should ask yourself when selecting a data source:

1. What data points are necessary to inform your analysis?
2. Do I already have access to this data, or must I find a dataset from another source?
3. Where are reliable and verifiable sources of this data?
4. How often is the relevant data collected and updated?
5. How is the data licensed for use, and is there a cost?
6. Is the data in a format that I can use, or convert to use, with my tools?

**3.1.4 Lab - Combining Relevant Data in Excel**

**3.1.5 Static and Streaming Data**

There are **two types** of data that analysts work with: **static data and streaming** data. Data that is received and **stored prior to performing analysis** on the data is **considered static data**. When each event is **processed and analyzed as it is received** and subsequent results are used or stored, the data is **referred to as streaming data**.

Consider data from a movie review site. *A data analyst wants to categorize each viewer comment by whether it is positive or negative during the week a new movie was previewed*. The analyst can treat the data as either static or streaming. If the data is going to be treated as static data, it will be stored in the database for later analysis at the end of the week. If it is going to be treated as streaming data, each entry in the comments will be analyzed as it is observed, in real-time. Often, the **results of analysis of streaming data are represented as a continuously updated dashboard, or as input to an automated function.**

**3.1.7 Data Types and Formats**

After data has been accessed from different sources, it requires preparation for analysis. In fact, experts in the field of Data Science estimate that **data preparation can take up 50 to 80 percent of the time required to complete an analysis.**

The **data sets** to be analyzed **may come from diverse sources** that may not necessarily be compatible with each other. **Another** potential data preparation **issue** is **data** that **will need to be converted** to a different type **to be used for statistical analysis**, *such as text representing dates that needs to be converted to numbers.* Data types matter when calculations and comparisons must be made using the data. Common data types are shown in the following table.

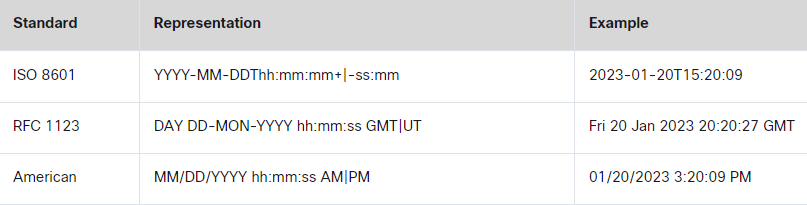
**Common Data Types**

|  |  |
| --- | --- |
| **Data Type** | **Description** |
| String | Data that is treated as text. It is composed of letters, numbers not to be used in computation, and symbols such as punctuation. Finally, string data typically includes white space, or the spaces used to separate and format text. |
| Integer | Whole numbers, or numbers that don’t include decimals. Depending on the computer language, integers may or may not include negative numbers. Integers are most often used to order or rank things. |
| Floating point | Numbers with decimal places. These numbers are frequently employed in statistical analysis. |
| Date and time | Important in recording when an observation in a data set was made. Date and time formats can vary widely between data sources. |
| Boolean | Data that is treated as either True or False. Typically, this is written as TRUE or FALSE to indicate a Boolean result instead of a string. |

In **addition to different data types**, a **single type of data can be formatted differently** in different data sets, depending on its source. For example, **different languages use different symbols** to represent the same word, and British **English uses different spellings of some words than American English**. An analysis of British and American English text searching for modes of travel, for example, would need to look for both **airplane** (American) and **aeroplane** (British) in order return accurate results.

**Time and data formats** also often present challenges. Although times and dates are very specific, they are represented in a wide variety of formats and must be converted to a single format for an analysis to have any value. For example, dates may be the day before the month in some countries, while other countries may present data with the month before the day. Similarly, time may be represented in 12-hour format with the AM and PM designation or could be represented in 24-hour format. Different time and date formats are shown in the following table.

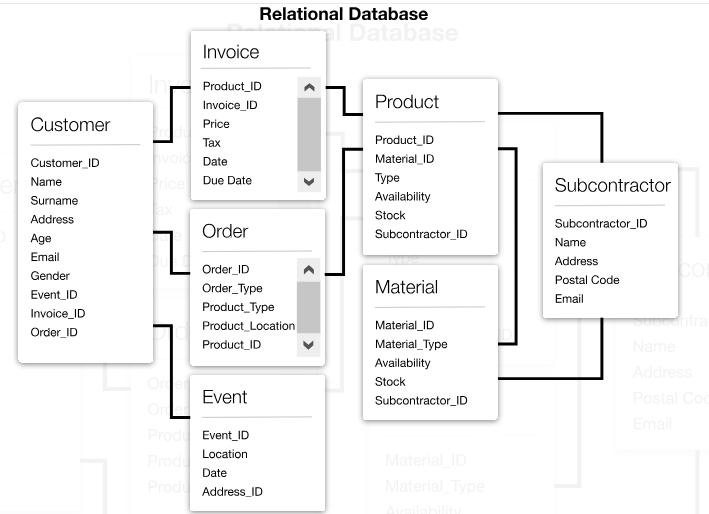
**Common Time and Date Formats**



**3.2 Data in Structured Files**

**3.2.1 Characteristics of Structured Data**

Structured data refers to data that is entered and maintained in defined fields within a file or record. Structured data is easily entered, classified, queried, and analyzed by a computer. This includes data found in relational databases and spreadsheets. For example, when you submit your name, address, and billing information to a website, you are creating structured data. The database may force you to enter it in a certain format for a computer to interpret it easily.



Some of the characteristics of structured data include:

* There is a well-defined and organized structure.
* It can be stored in tables, usually within vertical columns and horizontal rows.
* The content and format of the data is documented.
* It is organized into files, records, and fields.
* It can be searched, sorted, and queried.
* Input controls can reduce the possibility of invalid data.

**3.2.3 CSV Files**

Different applications create files in different formats that are not necessarily compatible with one another. For this reason, a universal file format is needed. Comma-separated values (CSV) files are a type of plain text file format that is widely used to standardize data. CSV files use commas, or other characters such as tabs, spaces, or colons, to separate columns in a table of data, and the newline character to separate rows, as shown in the following example.



Each row can also be referred to as a record. CSV files are commonly used for importing and exporting data to spreadsheets and traditional databases but can also be used as input for analytic programs and visualization tools.

JSON (JavaScript Object Notation) and XML (Extensible Markup Language) are also common, standardized plain text file types often used for representing data records. JSON is a lightweight data-interchange format that is easy for humans to read and write. XML is a markup language that is similar to HTML. These file formats are compatible with a wide range of applications. Converting data into a common format is a valuable way to combine data from different sources.

**3.2.5 Structured File Types**

There are many different types of structured data files that can either be created by humans or machine-generated. Here are a few examples of structured files:

**Relational Databases**

A relational database is a collection of tables with columns and rows that are connected by pre-defined relationships. These items are organized as a set of tables. Tables are used to hold information about the objects to be represented in the database. Each column in a table holds a certain kind of data, and each field in that column stores the actual value of an attribute.

**Logs**

Log files are a machine-generated historical record of everything and anything that happens within a system, such as transactions, errors, or intrusions. Log files are usually considered structured data, because they are machine-generated and so adhere to a standard format.

**Spreadsheets**

A spreadsheet file is an example of a flat file database. A flat file database stores records in a single file with no hierarchical structure. Spreadsheets are organized similarly to tables in a database, in that the data is organized into rows and columns, with related objects aligned either horizontally or vertically.

**Sensor readings**

Sensor output is usually collected in a standardized format, which may vary by manufacturer. Individual readings may be separated only by a delimiter or may be time dependent, such as 1 output per second, separated by timestamps.

**Transactional Records**

Records of transactions can be stored in many different formats, depending on the type of transaction and its source. Some transactions are entered manually into forms, while others can be machine generated.

**Lab - Working with CSV Files**

**3.3 Unstructured Data**

**3.3.1 Big Data and Decision Making**

Unstructured data lacks the organization found in structured data. Unstructured data is raw data, not organized in a predefined way. It does not possess a fixed schema that identifies the type of data or its format. This type of **data lacks a set way of entering or grouping the data**, and then analyzing the data.

Examples of unstructured data include the **content of photos, audio, video, web pages, blogs, books, journals, white papers, PowerPoint presentations, articles, email, wikis, word processing documents, and text in general.** Even the PDF version of this chapter is unstructured. The text is searchable, but it is not organized in a predefined format. Unstructured data can even be a traffic camera feed that is continuously sending images for processing.

**Both structured and unstructured data are valuable** to individuals, organizations, industries, and governments. It is important for organizations to take all forms of data and determine ways to format that data so it can be managed and analyzed.

**3.3.5 Sources of Unstructured Data**

There are many sources of unstructured data on the internet, in the form of text, images, videos, and audio. Public web forums and blogs also generate data. Social media such as YouTube, Facebook, instant messaging, RSS(An RSS document includes full or summarized text, and metadata like publishing date and author's name.), and Twitter all add to the data found on the internet. Most of this data is unstructured, which means it is not easy to categorize into a database without some type of processing. Data that is unstructured can be extracted and processed in a number of ways:

**NoSQL Databases and Data Lakes**

Unstructured data is often stored in non-relational databases or in data lakes, which are centralized repositories for data obtained from IoT devices, web sites, mobile apps, social media and other sources of raw data. These types of repositories are used to store real-time data in its original format. We will learn about NoSQL later in the course.

**Web Scraping**

Web pages are created to provide data to humans, not machines. “Web scraping” tools automatically extract various forms of data from HTML pages. Typically, web scraping is an automated process which uses a bot or web crawler to gather and copy specific data from the web to a database or spreadsheet. The data can then be easily analyzed.

**Application Program Interfaces (APIs)**

Many large web service providers, such as Facebook, provide standardized interfaces to collect data from them automatically, using APIs. The most common approach is to use RESTful application program interfaces (APIs). RESTful APIs use HTTP as their communication protocol and JSON files to store the data. Internet websites like Google and Twitter gather large amounts of static and streaming data. APIs for these sites allow data analysts and engineers to access subsets of the large amounts of data they are constantly generating.

**3.4 Data Preparation**

**3.4.1 Why is Data Dirty?**

**3.4.3 ETL and ELT Processes**

ETL and ELT are two versions of the same process for moving data through a pipeline. They contain the same steps but in different orders for different use cases.

**Extract, Transform and Load (ETL)**is a process for collecting data from this variety of sources, transforming the data, and then loading the data into a database. One company’s data might be found in Word documents, spreadsheets, plain text, PowerPoints, emails and PDF files. Another company’s data may be housed in relational databases. This data can be stored in a variety of different formats, making it difficult to combine and analyze, so the transformation happens before loading.

In an **Extract, Load, Transform (ELT)** process, the load and transform steps are reversed. ELT enables raw data to skip the transformation step and go straight to storage in an unstructured form. Transformation then occurs on the stored data as it is used. The ELT process is used primarily for large amounts of unstructured data.

**Step 1. Extract**

In this step, data is located and gathered from various sources in order to be converted into a single format for analysis. The data may be extracted from a relational database, NoSQL, flat files, XML files, or other formats.

**Step 2. Transform**

Data usually must be transformed before it can be loaded into a data warehouse for analysis. The transform step uses rules to transform the source data to the type of data needed for the target database. This includes converting any measured data to the same dimension (e.g., Imperial to Metric). The transformation step also requires several additional tasks. Some of these tasks are joining data from several sources, aggregating, sorting, determining new values that are calculated from aggregated data, and then applying validation rules.

Data (possibly including some empty or error data) may go through another part of the transform step known as ‘cleaning’ or ‘scrubbing’ data, and validation lets you know whether the data needs cleaning. Some examples of data cleaning are removing blank records and standardizing formats such as date, time, and location. The cleaning part of the transform step further ensures the consistency of the source data.

**Step 3. Load**

The transformed data is then loaded into the database for querying. The actual load process varies widely, depending on the types of source data, the type of target database, and the type of querying that is to be done. Some organizations may also overwrite existing data with newer cumulative data. During the load step, rules that have been defined in the database schema are applied. These rules check for necessary characteristics like uniqueness and consistency of data or that mandatory fields aren’t empty. These rules help to ensure that the loading and any subsequent querying of the data is successful.

**3.4.5 Cleaning Data**

**Lab - Preparing Data**

