For example, the number of correct answers in a quiz would be discrete as there is finite and countable number of correct answers. But, time to complete a task is continuous since it could take any value (e.g., 15.35 minutes) as time forms an interval from 0 to infinity. Some other examples of discrete data are: number of trees in a garden, number of children in a family, number of database books in a library, number of languages a person can speaks. Similarly, weights of persons, height of children, time to go on bed, speed of the car are continuous data.

1.2.3 Nominal, Ordinal, Interval and Ratio Data

The term **'nominal'** originates from the Latin word 'nomen' which means 'name' and it basically refers to categorically discrete data such as name of your university, name of of a book, etc. Nominal items are usually categorical, in that they belong to a definable category and differentiated by a simple naming system. The only thing a nominal scale does is to say that items being measured have something in common, although this may not be described. Nominal items may have numbers assigned to them, e.g., to represent a person's gender 0 can be assigned to male and 1 can be assigned to female, but these assignments are used just to simplify the representation and referenceing and not to define any rank.

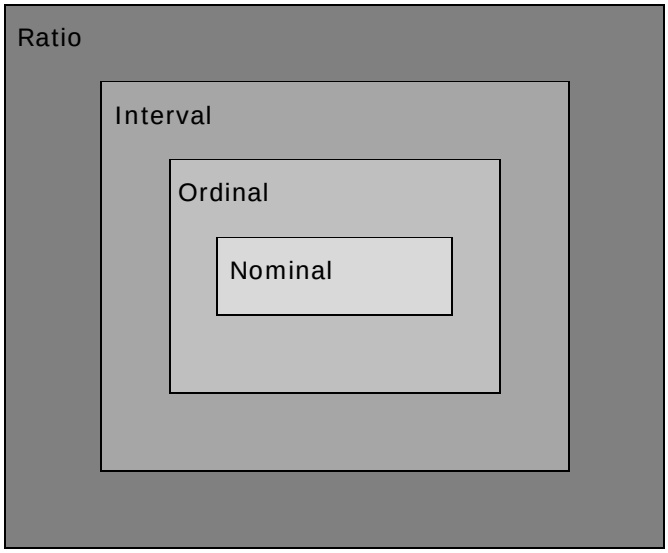

Figure 1.2 – Graded relationship among nominal, ordinal, interval, and ratio data

Like nominal data, **ordinal data** are usually categorical, in that they belong to a definable category. But, in contrast to nominal data, ordinal data have some natural ordering which is defined by their position on the scale. The positions may indicate temporal positions, superiority, etc. For example, the first, second and third person in a race, designations in an organization. The order of ordinal data is often defined by

#

assigning numbers to show their relative positions but the purpose is to show sequence only – we cannot do arithmetic with ordinal numbers. Instead of numbers, letters or other sequential symbols may also be used as appropriate.

Although ordinal data are ordered we can not state with certainty whether the intervals between each value are equal. For example, in a race competition the ordinal positions (1st, 2nd and 3rd) do not tell us anything about the absolute magnitude of the difference between 1st and 2nd or between 2nd and 3rd. That is, we know 1st was before 2nd, and 2nd was before 3rd, but we do not know how close 3rd was to 2nd or how close 2nd was to 1st. **Interval data** is like ordinal except we can say the intervals between each value are equally splited. Interval data are measured along a scale in which each position is equidistant from one another. This allows for the distance between two pairs to be equivalent in some way.When a variable is measured on an interval scale, the distance between numbers or units on the scale is equal over all levels of the scale. The most common example is temperature in degrees Fahrenheit. The difference between 15 and 25 degrees is the same magnitude as the difference between 75 and 85.

With interval scales, there is no absolute zero point. For this reason, it is inappropriate to express interval level measurements as ratios; it would not be appropriate to say that 40 degrees is twice as hot as 20 degrees. **Ratio data** is interval data with a natural zero point. For example, time is ratio data since 0 time is meaningful. Moreover, not only can we say that difference between two hours and four hours is the same as the difference between six hours and eight hours (equal intervals), but we can also say that eight hours is twice as long as four hours (a ratio comparison).

Figure 1.2 shows the graded relationship (as each one adding more to the next) among nominal, ordinal, interval, and ratio data. Ordinal data is also nominal; interval data is also ordinal which is in turn nominal, and so on.

1.2.4 Unstructured, Semi-Structured and Structured Data

Data can be embedded within unstructured, semi-structured or structured files and classified accordingly. Text files (created using word processors like Notepad, Wordpad, MS-Word, etc.) are generally unstructured in nature as they do not have any kind of annotation. Data stored in text files are termed as **unstructured data** and it is not a straightforward task for machine to interpret them correctly. For example, in a text file containing biography of an artist it a complex task for a machine to determine whether the numeric value 45 stored in it represents his/her house number or age. Text mining is a new approach to extract and analyze data stored in unstructured text documents.

#

Data stored in web pages are called **semi-structured data** as web pages are partially structured. For example, in a webpage, some part of it like title, body, etc. are annotated with markup language tags but an annotated part may have unstructured data. Hence, by applying simple pattern matching (using <title> and <body> tags) it is easy to identify the title and body of a web page, but it is very difficult to identify and extract data that are embedded within body – as it is unstructured. Web content mining deals with the problem of extracting and analyzing data stored in web pages.

Data stored in databases are called **structured data** as in a database every piece of data is clearly defined and marked. Data mining is a technique that deals with the problem of analyzing structured data.

1.3 File-Based Approach

In our society we always try to know about our predecessors for the purpose of adopting the features that were present in them and preventing us from repeating the same mistakes did by them. As the file-based system (also called traditional file system) is the predecessor of database system, we will not depart from this tradition. Although the file-based approach is largely obsolete, there are good reasons for studying it:

- Understanding the working of file-based system can assist us to understand the database system.
- Understanding the problems inherent in file-based systems may prevent us from repeating the same problems in database systems.
- Understanding of how file-based system works can be useful to migrate from a file-based system to a database system.

A **file** is simply a collection of **records**, where each record contains logically related data. File-based systems were an early attempt to computerize the manual filing system. For example, in an organization a manual file is set up to hold all external and internal correspondence relating to a project, product, task, client, or employee. Typically, there are many such files, and for safety they are labeled and stored in one or more cabinets. For security, the cabinets may have locks or may be located in secure areas of the building. In our own home, we probably have some sort of filing system which contains receipts, invoices, bank statements, and such like. When we need to look something up, we go to the filing system and search through the system starting from the first entry until we find what we want. Alternatively, we may have an indexing system that helps to locate what we want more quickly. For example, we may have divisions in the filing system or separate folders for different types of item that are in some way *logically*