**Data Warehouse**

**Data** is a collection of facts, such as numbers, words, measurements, observations or even just descriptions of things.

In computing, a **data warehouse**, also known as an **enterprise data warehouse (EDW)**, is a system used for reporting and data analysis, and is considered a core component of business intelligence. DWs are central repositories of integrated data from one or more disparate sources.

A data warehouse is a large collection of business data used to help an organization make decisions.

**Extract Transform Load (ETL)**

**Extract** is the process of reading data from multiple and different types of sources. In general, the extraction phase aims to convert the data into a single format appropriate for transformation processing.

**Transform** is data cleansing, which aims to pass only "proper" data to the target.

In other cases, one or more of the following transformation types may be required

* Selecting only certain columns to load: (or selecting null columns not to load). For example, if the source data has three columns (aka "attributes"), roll\_no, age, and salary, then the selection may take only roll\_no and salary. Or, the selection mechanism may ignore all those records where salary is not present (salary = null).
* Translating coded values: (*e.g.*, if the source system codes male as "1" and female as "2", but the warehouse codes male as "M" and female as "F")
* Encoding free-form values: (*e.g.*, mapping "Male" to "M")
* Deriving a new calculated value: (*e.g.*, sale\_amount = qty \* unit\_price)
* Sorting or ordering the data based on a list of columns to improve search performance
* Joining data from multiple sources (*e.g.*, lookup, merge) and deduplicating the data
* Aggregating (for example, rollup — summarizing multiple rows of data — total sales for each store, and for each region, etc.)
* Generating surrogate-key values
* Transposing or pivoting (turning multiple columns into multiple rows or vice versa)
* Splitting a column into multiple columns (*e.g.*, converting a comma-separated list, specified as a string in one column, into individual values in different columns)
* Disaggregating repeating columns
* Looking up and validating the relevant data from tables or referential files
* Applying any form of data validation; failed validation may result in a full rejection of the data, partial rejection, or no rejection at all, and thus none, some, or all of the data is handed over to the next step depending on the rule design and exception handling; many of the above transformations may result in exceptions, e.g., when a code translation parses an unknown code in the extracted data

**Load** is phase that loads the data into the end target, which can be any data store including a simple delimited flat file or a data warehouse.

**The Purpose of Dimensional Models**

The two primary goals for dimensional modeling are *ease of use* and *query performance*.

**What Is a Dimensional Model?**

A *dimensional model* is a data model organized for the purpose of user understandability and high performance. There are two basic parts of a dimensional model: the *dimensions* and the *facts*. These are the building blocks that comprise all dimensional models, simple or complex.

**Dimensions**

Dimensions are groupings of data elements in major business categories.

Common dimensions include the following:

* Customers
* Products
* Dates
* Suppliers
* Vendors
* Accounts

**Facts**

*Facts* are the measurement of business events. These are captured as specific information about a business event or transaction. These are measured, monitored, and tracked over time. Facts are typically the amounts and counts that show up as the body of reports. Facts are used as the basis for all calculations.

Examples of facts include units ordered, retail price, amount paid, claim payment amount, gross margin, budgeted dollars, revenue forecast, and loan balance, among others.

***Physical Database Design***

Because a dimensional model addresses query performance as one primary purpose of the design approach, not much is needed other than some physical decisions such as *physical partitioning* and *indexing* strategies.

**Business intelligence** is a broad all-encompassing term that is used to describe a wide variety of different types of data access. *Business intelligence* is the collection of one or more reports or analyses that provide insight into performance of a business organization.

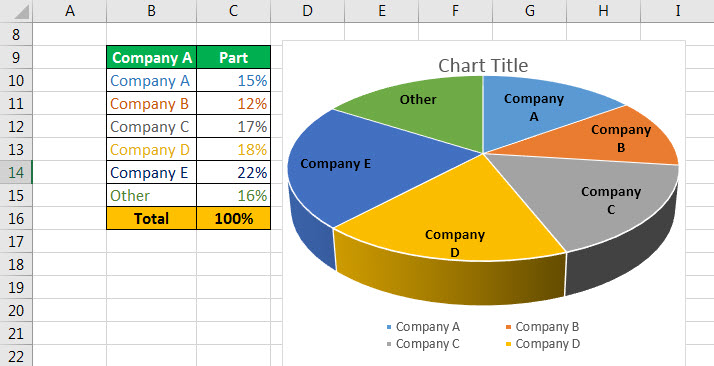
**Data quality** is ‘‘the state of completeness, validity, consistency, timeliness and accuracy that makes data appropriate for a specific use.’’

**Data validation** is the process of ensuring data have undergone data cleansing to ensure they have data quality, that is, that they are both correct and useful.

**Chart and Graph**

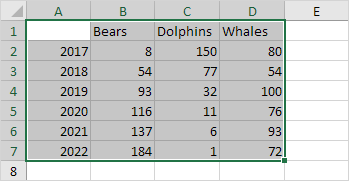
* **Pie charts** depict a portion of a whole. Now that might seem like trivial information, but you'd be amazed to know just how often this simple fact is misunderstood. You see, more often than not, people see a percentage as a data point and default to displaying that percentage in a pie chart. And 90% of the time, a pie chart is the best way to visualize a percentage. But again, remember that a pie chart depicts a part of a whole.
* **Bar graphs** are used to display trends grouped by categories or time. One axis is traditionally used to show the amount of something, while the other is used to label each bar.
* **Line graphs** are much like a bar graph, line graphs display trends over time or category. But they do this quite differently. When showing data over time, a line graph is often better than a bar chart. This is because trend lines are clearer, allowing you to see change over time far easier.
* **Pie or Doughnut chart**: Use this chart type to show proportions of a whole. Use it when the total of your numbers is 100%.

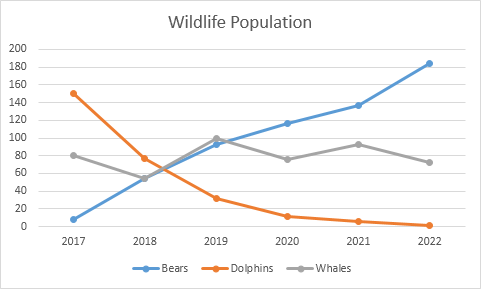
**Example:**



* **Line or Area chart**: Use this chart type to show trends over time (years, months, and days) or categories.

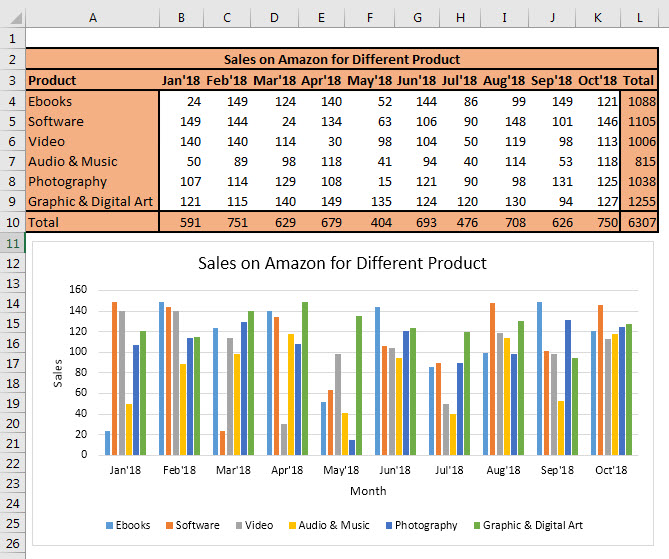
**Example:**





* **Column or Bar chart**: Use this chart type to visually compare values across a few categories.

**Example:**



* **Scatter (x, y) or Bubble chart**: Use this chart type to show the relationship between sets of values.

**Example:**

