**Identify data formats**

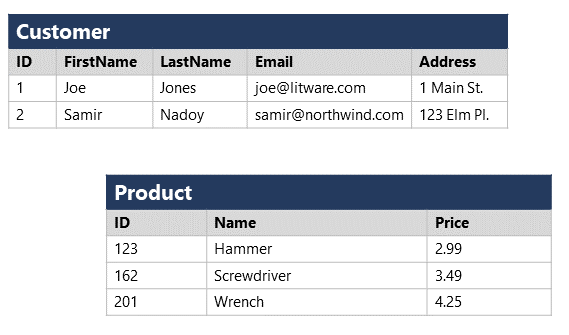
* 5 minutes

Data is a collection of facts such as numbers, descriptions, and observations used to record information. Data structures in which this data is organized often represents *entities* that are important to an organization (such as customers, products, sales orders, and so on). Each entity typically has one or more *attributes*, or characteristics (for example, a customer might have a name, an address, a phone number, and so on).

You can classify data as *structured*, *semi-structured*, or *unstructured*.

**Structured data**

Structured data is data that adheres to a fixed *schema*, so all of the data has the same fields or properties. Most commonly, the schema for structured data entities is *tabular* - in other words, the data is represented in one or more tables that consist of rows to represent each instance of a data entity, and columns to represent attributes of the entity. For example, the following image shows tabular data representations for *Customer* and *Product* entities.



Structured data is often stored in a database in which multiple tables can reference one another by using key values in a *relational* model; which we'll explore in more depth later.

**Semi-structured data**

*Semi-structured* data is information that has some structure, but which allows for some variation between entity instances. For example, while most customers may have an email address, some might have multiple email addresses, and some might have none at all.

One common format for semi-structured data is *JavaScript Object Notation* (JSON). The example below shows a pair of JSON documents that represent customer information. Each customer document includes address and contact information, but the specific fields vary between customers.

JSONCopy

// Customer 1

{

"firstName": "Joe",

"lastName": "Jones",

"address":

{

"streetAddress": "1 Main St.",

"city": "New York",

"state": "NY",

"postalCode": "10099"

},

"contact":

[

{

"type": "home",

"number": "555 123-1234"

},

{

"type": "email",

"address": "joe@litware.com"

}

]

}

// Customer 2

{

"firstName": "Samir",

"lastName": "Nadoy",

"address":

{

"streetAddress": "123 Elm Pl.",

"unit": "500",

"city": "Seattle",

"state": "WA",

"postalCode": "98999"

},

"contact":

[

{

"type": "email",

"address": "samir@northwind.com"

}

]

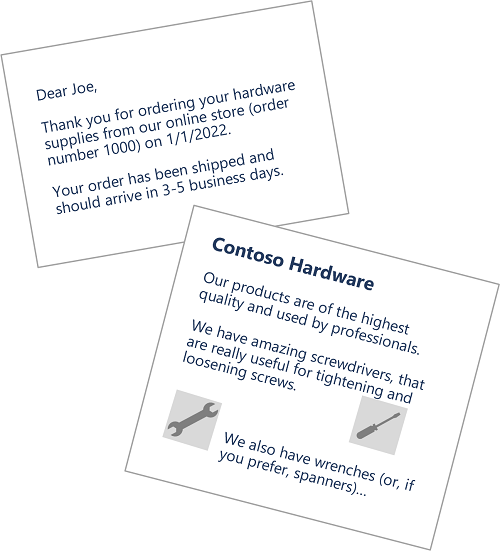
}

**Note**

JSON is just one of many ways in which semi-structured data can be represented. The point here is not to provide a detailed examination of JSON syntax, but rather to illustrate the flexible nature of semi-structured data representations.

**Unstructured data**

Not all data is structured or even semi-structured. For example, documents, images, audio and video data, and binary files might not have a specific structure. This kind of data is referred to as *unstructured* data.



**Data stores**

Organizations typically store data in structured, semi-structured, or unstructured format to record details of entities (for example, customers and products), specific events (such as sales transactions), or other information in documents, images, and other formats. The stored data can then be retrieved for analysis and reporting later.

There are two broad categories of data store in common use:

* File stores
* Databases

We'll explore both of these types of data store in subsequent topics

**Xác định định dạng dữ liệu**

Dữ liệu là một tập hợp các sự kiện như số, mô tả và quan sát được sử dụng để ghi lại thông tin. Các cấu trúc dữ liệu nơi dữ liệu này được tổ chức thường đại diện cho các **thực thể** quan trọng đối với một tổ chức (chẳng hạn như khách hàng, sản phẩm, đơn đặt hàng, v.v.). Mỗi thực thể thường có một hoặc nhiều **thuộc tính** hoặc đặc điểm (ví dụ: khách hàng có thể có tên, địa chỉ, số điện thoại, v.v.).

Bạn có thể phân loại dữ liệu thành **có cấu trúc**, **bán cấu trúc** hoặc **không cấu trúc**.

**Dữ liệu có cấu trúc (Structured data)**

Dữ liệu có cấu trúc là dữ liệu tuân theo một **sơ đồ** cố định, vì vậy tất cả dữ liệu đều có các trường hoặc thuộc tính giống nhau. Thông thường nhất, sơ đồ cho các thực thể dữ liệu có cấu trúc là **dạng bảng** - nói cách khác, dữ liệu được biểu diễn trong một hoặc nhiều bảng bao gồm các hàng để biểu thị mỗi thể hiện của một thực thể dữ liệu và các cột để biểu thị các thuộc tính của thực thể. Ví dụ, hình ảnh sau đây cho thấy biểu diễn dữ liệu dạng bảng cho các thực thể **Khách hàng** và **Sản phẩm**.

[Ảnh minh họa biểu diễn dữ liệu dạng bảng cho các thực thể Khách hàng và Sản phẩm]

Dữ liệu có cấu trúc thường được lưu trữ trong một cơ sở dữ liệu trong đó nhiều bảng có thể tham chiếu lẫn nhau bằng cách sử dụng các giá trị khóa trong một **mô hình quan hệ** (chúng ta sẽ khám phá chi tiết hơn về mô hình này sau).

**Dữ liệu bán cấu trúc (Semi-structured data)**

Dữ liệu bán cấu trúc là thông tin có một số cấu trúc, nhưng cho phép có một số biến thể giữa các thể hiện của thực thể. Ví dụ, trong khi hầu hết khách hàng có thể có địa chỉ email, một số khách hàng có thể có nhiều địa chỉ email và một số khác có thể không có địa chỉ email nào.

Một định dạng phổ biến cho dữ liệu bán cấu trúc là **JavaScript Object Notation (JSON)**. Ví dụ dưới đây cho thấy một cặp tài liệu JSON đại diện cho thông tin khách hàng. Mỗi tài liệu khách hàng bao gồm thông tin địa chỉ và liên hệ, nhưng các trường cụ thể khác nhau giữa các khách hàng.

JSON

// Khách hàng 1

{

"firstName": "Nam",

"lastName": "Nguyen",

"address":

{

"streetAddress": "Số 1 phố chính",

"city": "Hà Nội",

"state": "Hà Nội",

"postalCode": "100000"

},

"contact":

[

{

"type": "nhà riêng",

"number": "024 3856 1234"

},

{

"type": "email",

"address": "nam.nguyen@congty.com"

}

]

}

// Khách hàng 2

{

"firstName": "Hoa",

"lastName": "Tran",

"address":

{

"streetAddress": "Số 123 ngõ Elm",

"unit": "500",

"city": "Thành phố Hồ Chí Minh",

"state": "TP. Hồ Chí Minh",

"postalCode": "700000"

},

"contact":

[

{

"type": "email",

"address": "hoa.tran@congtykhac.com"

}

]

}

Use code [with caution.](https://gemini.google.com/faq#coding)

content\_copy

**Lưu ý**

JSON chỉ là một trong nhiều cách để biểu diễn dữ liệu bán cấu trúc. Mục đích ở đây không phải là cung cấp một bài kiểm tra chi tiết về cú pháp JSON, mà là để minh họa bản chất linh hoạt của các biểu diễn dữ liệu bán cấu trúc.

**Dữ liệu không cấu trúc (Unstructured data)**

Không phải tất cả dữ liệu đều có cấu trúc hoặc thậm chí là bán cấu trúc. Ví dụ: tài liệu, hình ảnh, dữ liệu âm thanh và video và các tệp nhị phân có thể không có cấu trúc cụ thể. Loại dữ liệu này được gọi là dữ liệu không cấu trúc.

**Kho lưu trữ dữ liệu (Data stores)**

Các tổ chức thường lưu trữ dữ liệu ở định dạng có cấu trúc, bán cấu trúc hoặc không cấu trúc để ghi lại chi tiết của các thực thể (ví dụ: khách hàng và sản phẩm), các sự kiện cụ