Traditionally, when we store data in a database table, we define one or more indices that have unique value across the tables. For example, we could have a table with social security number and employees and the rest of the dataset has information about date of hire, salary, region, etc. Another table could have the same indices, but he information in this table is related to perhaps medical history. With indexing we can 'link' tables together and augment the information the exists in those tables.

In some cases the table is index on something that does not a have a unique value (like social security), but can have a finite number of values such as 'Male','Female','Other'. In this case we can have multiple records with these values as shown below. Now, let's say we have a second index, 'Country'; which can have a finite number of values.

	Gender	Age	Country	TrickAndTreating
0	Male	22	Canada	No
1	Male	45	usa	No
2	Female	48	US	No
3	Male	57	usa	No
4	Male	42	USA	Yes

If we index on Gender and Country, we get this:

2]:

Gender	Country	Age	TrickAndTreating
Male	United States	34	No
	A tropical island south of the equator	Old enough to know better	No
	America	48	No
	America	36	Yes
Female	America	13	Yes
	America	47	No
	Australia	66	No
	Australia	45	Yes
I'd rather not say	Australia	over 40	No
Other	Australia	42	No
Male	Austria	33	No
	Brasil	34	No
Female	Canada	44	No
	Canada	37	No
	Canada	38	No
	Canada	37	No
	Canada	58	No
	Canada	64	No
	Canada	37	No
	Canada	33	No

This is called Hierarchical indexing. With this method, we can index in as many levels as it makes sense. For example, we could index further with setting 'TrickAndTreating' as the 3rd index. We can then sort on Gender(level=1), Country(level=2), or TrickAndTreating'(level=3).

Hierarchical indexing is a powerful tool in data wrangling. In Python and pandas, there are numerous techniques and features we can use to manipulate and extract data from and otherwise messy set od data.

References

1. Hierarchical Indexing - https://jakevdp.github.io/PythonDataScienceHandbook/03.05-hierarchical-indexing.html