Multiple Tables in Pandas

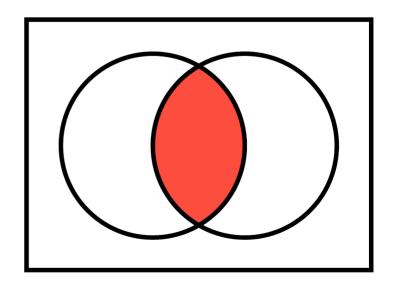code|cademy

Efficient Data Storage with Multiple Tables

For efficient data storage, related information is often spread across multiple tables of a database. Consider an e-commerce business that tracks the products that have been ordered from its website. Business data for the company could be split into three tables:

- `orders` would contain the information necessary to describe an order: `order_id`, `customer_id`, `product_id`, `quantity`, and `timestamp`
- `products` would contain the information to describe each product: `product_id`, `product_description` and `product_price`
- `customers` would contain the information for each customer: `customer_id`, `customer_name`, `customer_address`, and `customer_phone_number`

This table structure prevents the storage of redundant information, given that each customer's and product's information is only stored once, rather than each time a customer places an order for another item.

Pandas DataFrame Inner Merge

In Pandas the `.merge()` function uses an inner merge by default. An inner merge can be thought of as the intersection between two (or more) DataFrames. This is similar to a Venn diagram. In other words, an inner merge only returns rows both tables have in common. Any rows in one DataFrame that are not in the other, will not be in the result.