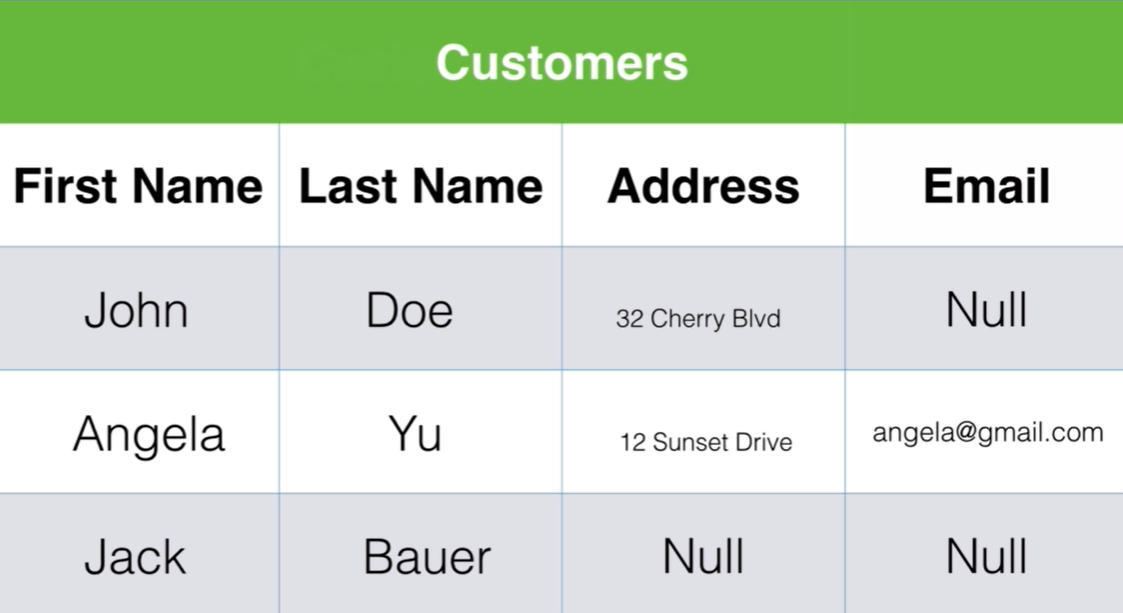
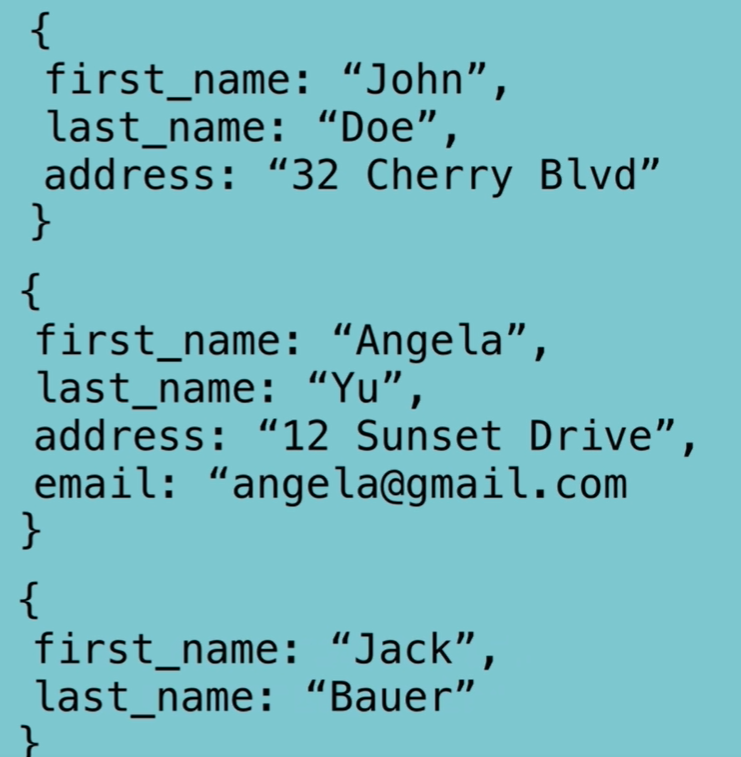
SQL vs. NoSQL

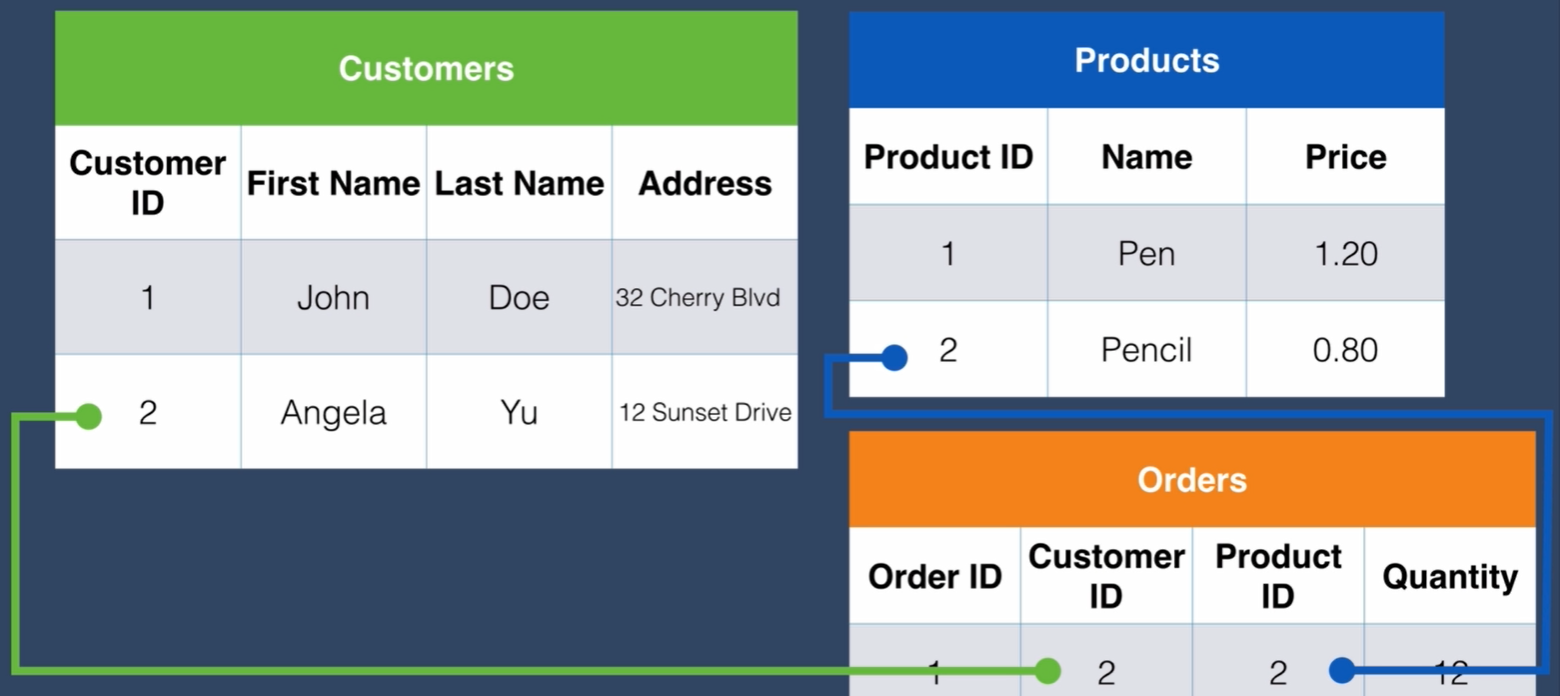


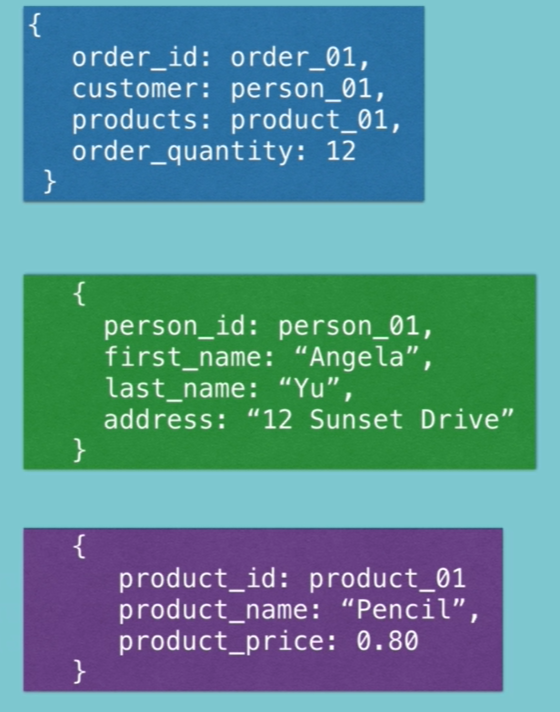


* Structure Difference
  + SQL uses table representations with fixed columns and null values for empty spaces
    - 
  + NoSQL: If you were working with Mongo the same data would be represented as JSON objects. So, you would have key-value pairs such as first name, Last name, and address and if in one of those documents you had an extra key-value pair such as email then it doesn't affect any of the other data records. And none of these records or documents actually have to be the same shape or follow the same structure.
    - 

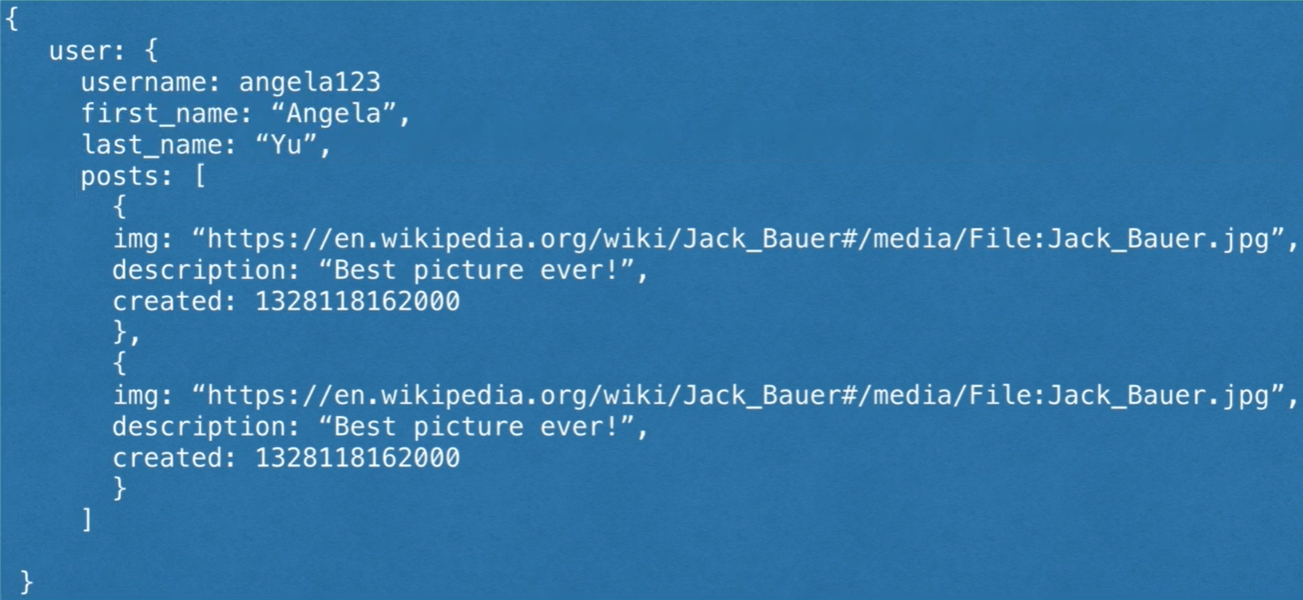
SQL vs NoSQL continued:

* Modernity
  + NoSQL databases tend to be flashier, newer with modern syntax and modern methods
  + but SQL database is old and reliable and it likes structure.
* Relationships
  + SQL databases of really good at establishing relationships between your data.



* + No SQL: is a bit clumsy with regards to relationships
    - 
* Scalability:
  + SQL:- It gets slower and slower the more rows of data you add until a point where the file is so big that your computer can no longer handle it.
  + NoSQL:- It is more scalable

When it is best to use each: SQL vs NoSQL?

* SQL: - orders, customer details, products inventory, things that have lots of relationships with each other.
* NoSQL: - if you have a website where you have something that's more of a one to many kinds of relationship like here for example say if we were making a database for Instagram, then we might have a user name, a first name, last name, and then we might have a post array. And so, this is the array of objects and those objects each have an image URL description and the date that they were created. In this case where you have a single user generating lots of content creating a one user to many posts’ relationship, then this is much easier to map out using something like MongoDB.
  + 

Summary of Differences

