Multiple JOINs

* Takeaway Tips for Multiple JOINs:
  + It’s important to imagine what you are creating at each JOIN – think about the rows that might be lost or gained at every JOIN
  + Although not a hard-and-fast rule, it’s generally a good idea to star with larger tables. If you do any type of JOIN with the “vet” table on the left, it cannot grow beyond its existing rows unless you use RIGHT or OUTTER JOINs, and both of these aren’t commonly used (some programs don’t even support RIGHT or OUTTER JOINs anymore)
  + Just because you get an answer, doesn’t mean it’s correct. This is why it’s so important to understand what’s happening with each subsequent JOIN.
  + SQL syntax is only half the battle when it comes to queries. In all our example queries, we had perfect syntax but we were getting the wrong answer. Understanding the data in the database is the other half of the equation. If we don’t know, for example, that the “vet” table only contained animals who had been to the vet, we would never have gotten the right answer.
  + Here’s another example of that: Imagine an address column that lists the state in which a person lives. If a person can enter whatever they want, you might get different responses that mean the same thing (Penn, Pa, PA, pa, PA, etc., for Pennsylvania). If you’re looking for all customers in Pennsylvania and you don’t have an OR or IN statement in your WHERE clause, you won’t get the right answer. SQL doesn’t know that all of these mean Pennsylvania.