Entity Relationship Diagram (ERD) Tutorial – Part 1 [1]

1. Hi, my name is Taylor and I'm from *Lucidchart*. Today you'll be learning all about *entity relationship diagrams* or ERD's. We're going to start off by discussing a high-level overview, and then together we'll dive in and build an example together, complete with entities, attributes, relationships, and cardinality. By the end of this video, you'll be able to build an entire entity relationship diagram from scratch.

2. Have you ever wondered why you get an error message telling you that your ideal *Twitter* handle is already taken? Or how *Amazon* can keep track of so many different orders and customers and products? The answers to these questions lie[2] within the creation of a ***database*** which, in other words, is a collection of information that is organized so data can be easily stored, managed, updated, and retrieved. Now there's a lot of moving information in a database, and understanding how the many elements of a database interact with each other can be difficult to grasp.

3. Engineers need a visual way to understand how all these separate elements are related to each other and how they're working together. To show this, they build ***entity relationship diagrams***. So let's talk about how you're going to make these ER diagrams. You can draw them out with pen and paper but it's going to be way easier for you to use a diagramming tool.

4. Today I'm gonna use an easy-to-use tool called ***Lucidchart***. And you can too, for free actually. By clicking on the top right corner here, you can access *Lucidchart*'s website, enter your email address, and have a free account in just a few seconds. That way you can follow along with me and continue building your own ER diagrams as well.

5. Before we can create an actual ERD, we need to better understand the individual components of the entity relationship diagram. And this is going to start with ***entities***, which are an object such as a person, place, or thing, to be tracked in the database. For example, in the case of buying something on *Amazon*, let's say a *Snuggy*, an entity could be a customer, an order, and lastly we can't forget our *Snuggies*, the product.

---

1 https://www.youtube.com/watch?v=QpdhBUYk7Kk
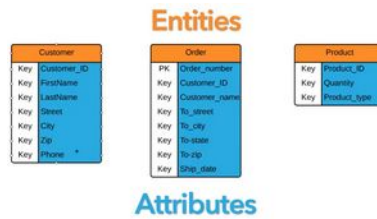2 Note: The locutor prononces "lies".

6. Now each one of these entities is going to have what we call **attributes**, which are various properties or traits. In this case, under the customer entity, we have a customer ID, first name, last name, street, city, zip and phone. It's important to remember that the entities in your database will be the rows, and that the attributes in your database will be depicted as the columns.

7. Now we have the different entities and the different attributes on the screen here, but now let's talk about the **relationships** that exist between these different entities. The relationships describe how these entities will interact with each other, if at all. And you do that by drawing a line in between them. So when I draw a line in between these particular elements, I'm showing that there is some sort of interaction or connection in some way.
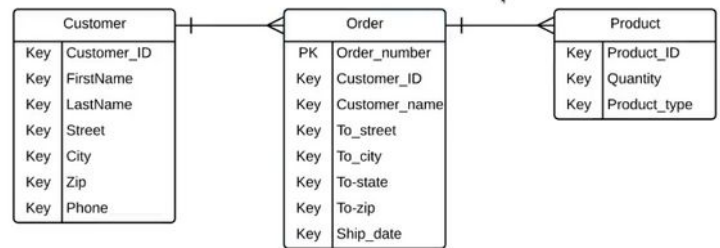
8. Now that we have our relationships in place, you'll see some funky notation attached to these lines. This in fact is called the **cardinality**, which further defines the relationship in a numerical context – particularly within minimums and maximums. So for example here on this right side, you can see some different types of cardinality we have in ER diagrams: we have *one*, *many*, *one and only one*, *zero or one*, *one or many*, *zero or many*. Now don't be afraid if this isn't making sense quite yet, we're gonna walk through some examples that'll help you understand this perfectly.
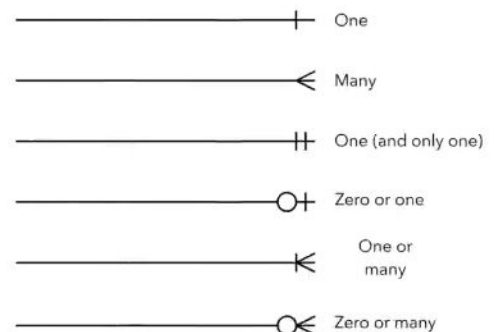
9. So let's talk about the relationship and the cardinality that exists **between a customer and the order**. Now the best way to do this is to think about it logically. We have to think about what is the minimum number of orders that a customer could have, and what is the maximum number of orders that a customer could have.

   • So let's start with **minimums**. What is the minimum number of orders that a customer could have? Well a customer could exist, but he or she could have *zero* orders. So to show that over here, we'll have that *zero* sign.

   • Now we have to think about the **maximum**. What are the maximum number of orders that a customer could have? Well, as you probably already know, a customer can have infinite orders. In the case of *Snuggies*, you can never just have one *Snuggy*. So to show that, we'll use this *zero or many* crow's foot notation.
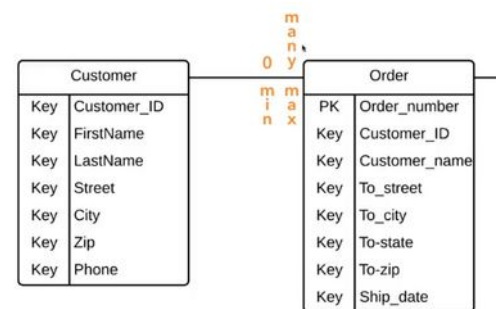
10. Now let's talk about the minimum and maximum relationship **between orders and customers**. So we ask ourselves the same questions: what is the minimum amount of customers that an order may have, and what is the maximum number of customers that an order may have?

    Now a specific order can only have *one and only one* customer. Well you can imagine the confusion that could come if the same specific order had lots of customers... So in this case, there can be only *one and only one* number of customers to an order, and we can show that using this sign here.
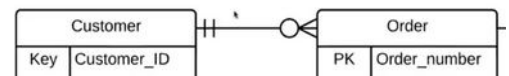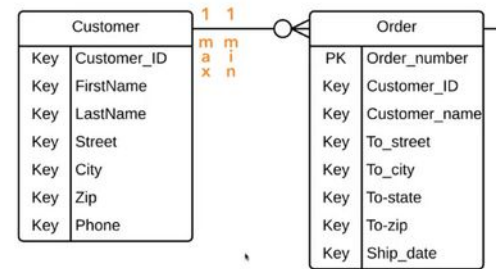
11. So now let's talk about the relationship or cardinality **between orders and products**. So we'll ask ourselves the same question: a certain order can have how many products? Well, for a order to exist, it has to have one product, but a lot of different products can be comprised of that order. So to show that on our diagram, we'll change this notation.

    Now we ask ourselves that question **in reverse**: a product can be a part of how many orders? Well, a product could be a part of no orders, but it also could be a product of many orders. So we're going to use this *zero or many* notation on our diagram.
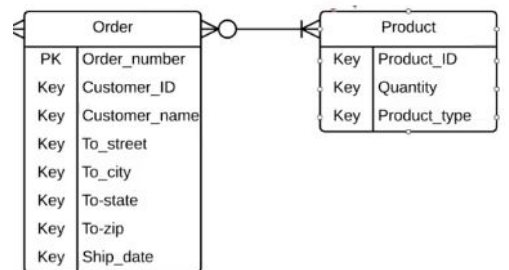
12. So now you've built your entire ER diagram, and we've built something small here, but now you have the **foundation and framework** that you need to build complex, complete entity relationship diagrams. Although this can seem a little bit overwhelming at first, just be sure to walk yourself through that logic and you'll be able to build entire complex ERD's. You'll also see on this diagram that there are some unfamiliar objects such as PK and FK, which refer to *primary keys* and *foreign keys* – something that we're going to be covering and the video coming soon, so stay tuned.
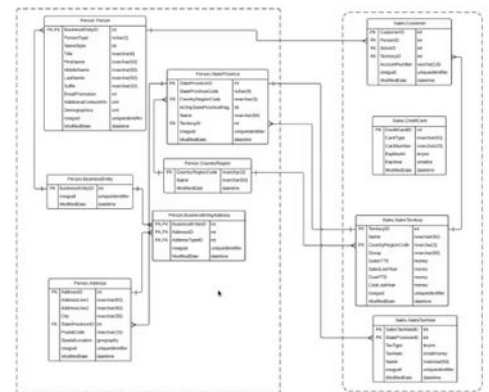
13. Additionally, if you need to actually have this diagram to be database-ready, you can use the **import and export features** of *Lucidchart* to have all of that done automatically for you. Using the *export* feature, you'll have that code automatically generated for you and exported to the database management system you're using.
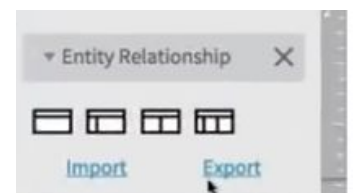
14. Thanks for watching this tutorial on ER diagrams. Subscribe to our channel below to access to more helpful videos and tutorials. Leave a comment as well if you have any thoughts or questions on ERD and don't forget: sign up for a free *Lucidchart* account by clicking on the link in the top right corner. Thanks!