

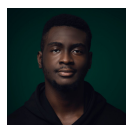
[Forum](#)

[Donate](#)

Learn to code – [free 3,000-hour curriculum](#)

JUNE 6, 2022 / [#DATABASE](#)

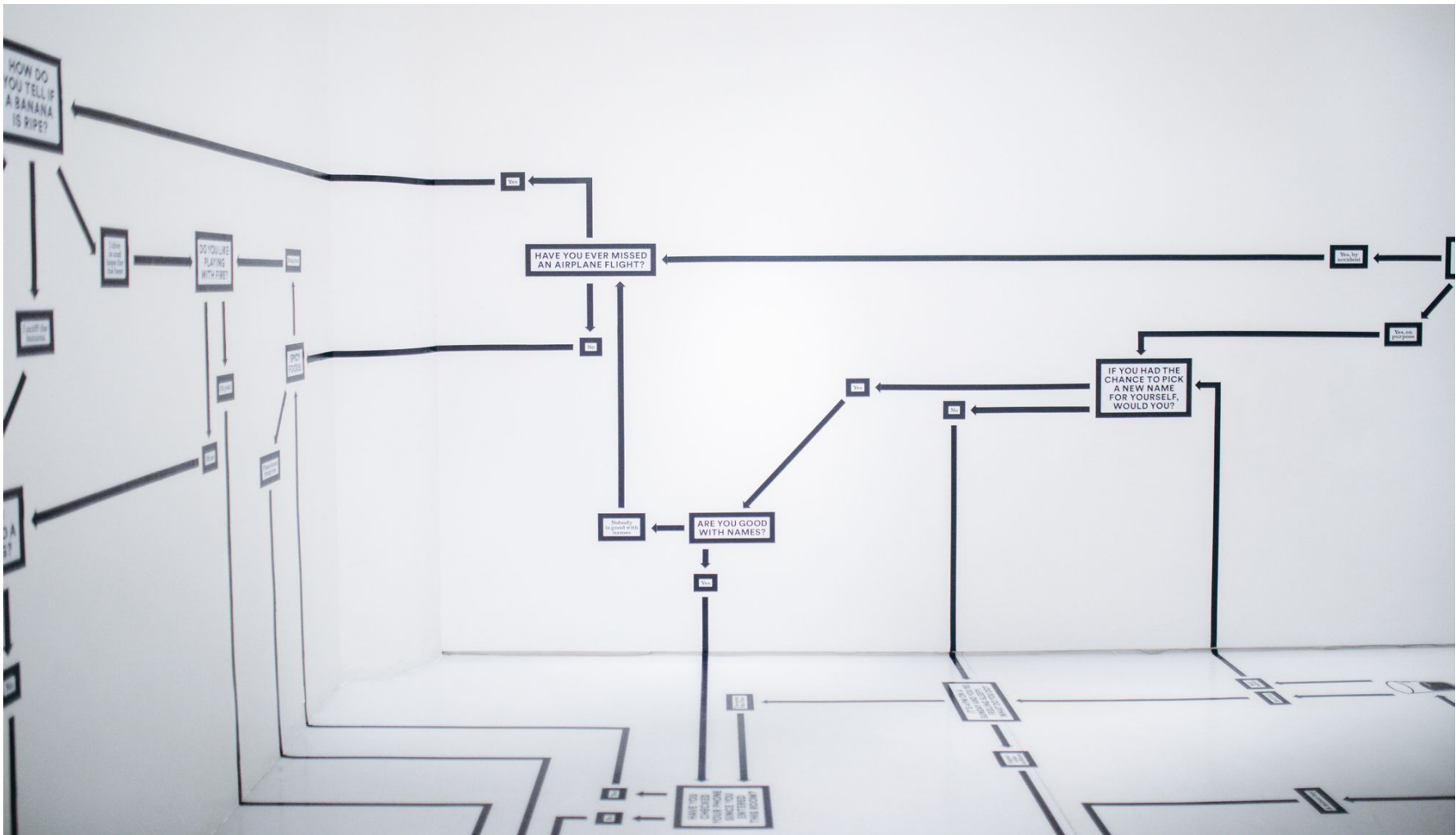
# Crow's Foot Notation – Relationship Symbols And How to Read Diagrams



Ihechikara Vincent Abba

[Forum](#)[Donate](#)

Learn to code – [free 3,000-hour curriculum](#)



[Forum](#)[Donate](#)[Learn to code – free 3,000-hour curriculum](#)

ADVERTISEMENT

Entity relationship diagrams (ERD) help us understand the connection between various "entities" that make up a system.

In software development, ERDs are mostly used in database design. This lets us create graphical representations of the entities that make up systems such as a database (you will understand this better with the examples in this tutorial).

In order to understand the relationship between entities in an ERD, we use specific symbols and notations.

Although there are various notations for understanding ERDs, we'll focus on Crow's Foot Notation which is one of the most commonly used when creating/designing ERDs.

[Forum](#)[Donate](#)

[Learn to code – free 3,000-hour curriculum](#)

various symbols in crow's foot notation that you can use to define the relationship between entities, and how to read and understand diagrams.

At the end of the tutorial, you should be able to understand and read diagrams, and create your own entity relationship diagrams making use of crow's foot notation to define your entity relationships.

You'll see the words: notation, indicators, and symbols used interchangeably.

## What Is an Entity in ERDs?

Before we look at some examples, let's talk about some of the key terms/components that'll make up the entity relationship diagrams we'll be working with.

The first is the **entity**. An entity simply represents an

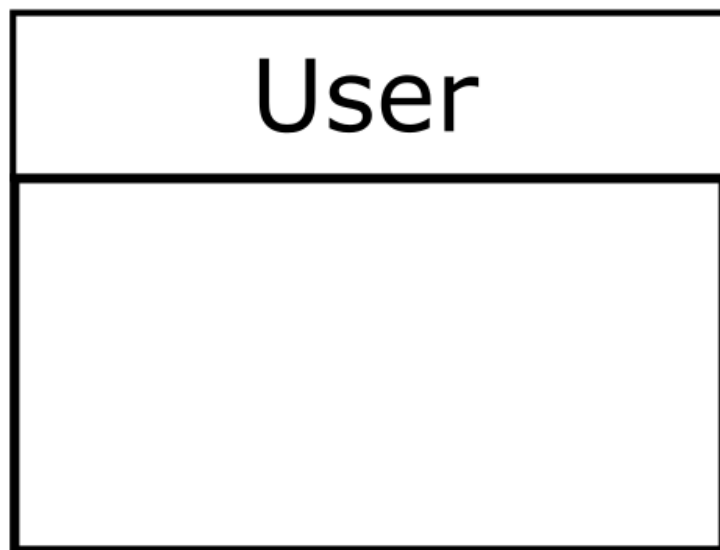
[Forum](#)[Donate](#)

Learn to code – [free 3,000-hour curriculum](#)

Note that the name of every entity should be singular (user) and not plural (users).

Here is what an entity looks like:

ADVERTISEMENT



[Forum](#)[Donate](#)

**Learn to code — [free 3,000-hour curriculum](#)**

The diagram above shows an entity called user. This entity will have information about the various users registered on a platform.

In the next section, we'll talk about attributes.

## What Are Attributes in ERDs?

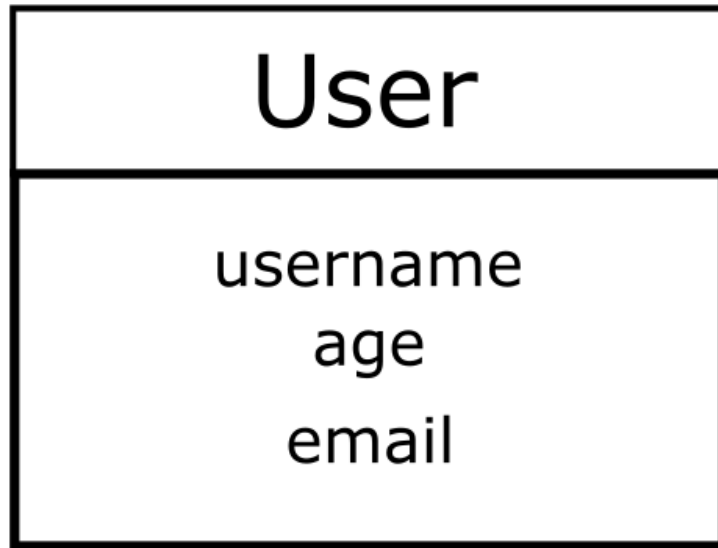
We've talked about entities and we know they store some sort of information about the object they're representing.

The information about an object is the attributes. So we can say the properties of an entity are the attributes.

Let's represent this using a diagram.

[Forum](#)[Donate](#)

Learn to code – [free 3,000-hour curriculum](#)



ADVERTISEMENT

The entity in the diagram above has three attributes – username, age, and email.

Now you have a clearer picture of what an entity is and its attributes.

[Forum](#)[Donate](#)[Learn to code – free 3,000-hour curriculum](#)

and email) which are referred to as the entity's attributes.

## Relationship Between Entities in ERDs

In the previous sections, we talked about entities and their attributes. In most cases, databases are made up of more than one entity.

To understand the relationship between one entity and another, we use lines to connect them. But these lines have notations (indicators) on them to specify the type of relationship that exists between two entities.

We'll use crow's foot notation to specify our entity relationships.



[Forum](#)[Donate](#)

[Learn to code – free 3,000-hour curriculum](#)

Before we see diagrams of the symbols associated with crow's foot notation, we need to discuss a key term in crow's foot notation.

One of the most important terms to know when using crow's foot notation is **cardinality**.

**Cardinality** acts as a parameter for the relationship between entities. For one entity, there is a minimum and maximum number that helps define its relationship with another entity.

Don't worry if these explanations seem confusing. As we go further, you'll understand them perfectly.

ADVERTISEMENT

Here are the symbols associated with the crow's foot notation:

[Forum](#)[Donate](#)

Learn to code – [free 3,000-hour curriculum](#)



The symbol/diagram above denotes zero in crow's foot notation. We know this because the of the zero/circle indicator at the right side of the horizontal line.

## One



The diagram above shows a horizontal line with a short vertical lines crossing it. The vertical line acts as the indicator – it denotes one.

[Forum](#)[Donate](#)

Learn to code – [free 3,000-hour curriculum](#)



The diagram above denotes many. You can easily remember this symbol because it looks like a crow's foot.

The three diagrams above are the basic representation of indicators in crow's foot notation. But in most cases, these indicators are combined to fully understand the relationship between entities.

By the time we start looking at some practical examples, you'd understand the meaning of these symbols better.

Before that, let's take a look at more diagrams and what they mean. We won't be introducing anything new – just a combination of the diagrams above.

ADVERTISEMENT

[Forum](#)[Donate](#)

Learn to code – [free 3,000-hour curriculum](#)



As can be seen above, the **zero or many** symbol/indicator in crow's foot notation is a combination of the zero and many indicators.

## One or Many



As expected, the **one or many** indicator is a combination of two indicators – one and many.

## One and only one

[Forum](#)[Donate](#)[Learn to code – free 3,000-hour curriculum](#)

The **one and only one** indicator has two "one" indicators. In our examples in the next section, you'll understand its use better.

## How to Use Crow's Foot Notation in Entity Relationship Diagrams

In the last section, we focused on crow's foot notation diagrams and what they mean. They serve as indicators that explain the relationship between one entity and another.

In this section, we'll dive in and look at some practical examples – this will help you fully understand how to use the crow's foot notation

[Forum](#)[Donate](#)

[Learn to code – free 3,000-hour curriculum](#)

sections, then some aspects of the diagrams we'll use in this section should be clear to you.

ADVERTISEMENT

## Crow's Foot Notation Example #1

In this example, we'll start with an assumption, create entities and denote their relationship using crow's foot notation.

We'll break this example into steps with diagrams leading to the final sketch.

### Step #1 - Our Assumption and Entities

Let's assume we have two entities in our database. A teacher and course entity. Here's a representation of that:

[Forum](#)[Donate](#)[Learn to code – free 3,000-hour curriculum](#)

## Step #2 - Relationship of the Teacher Entity and the Course Entity

Since we're the ones creating this model of the database, we get to set the rules! So for every teacher, they can only teach one course.

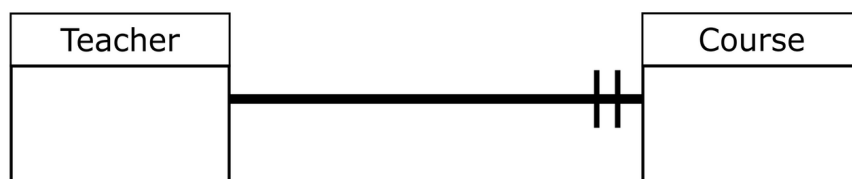
We'll assume this is a platform where users are taught programming languages. Each teacher can only teach one programming language.

The notation here will be **one and only one**. The notation will be placed on the right side of the horizontal line.

UML Class Diagram: Teacher and Course Relationship

[Forum](#)[Donate](#)

Learn to code – [free 3,000-hour curriculum](#)



ADVERTISEMENT

Remember when we talked about **cardinality**? Well, this is the perfect place to see it in practice. The minimum number of courses a teacher can take up is one, and the maximum is also one.

### Step #3 - Relationship of the Course Entity and the Teacher Entity

For each course, we want to have one or many teachers to choose from – meaning that one course can be taught by one or many teachers. The minimum here will be one



[Forum](#)[Donate](#)

Learn to code – [free 3,000-hour curriculum](#)

so the user can have one or many JavaScript teachers to learn from, one or many Python teachers to learn from, and so on.

The notation to be used is **one or many**. The notation will be placed on the left side of the horizontal line.

Here is the ERD:



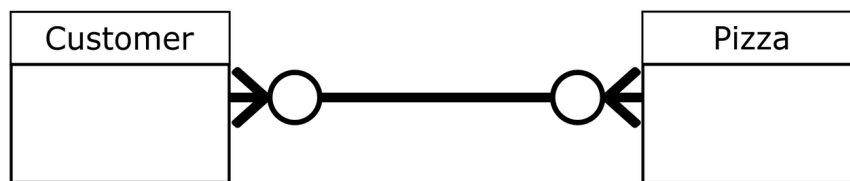
## Crow's Foot Notation Example #2

[Forum](#)[Donate](#)

**Learn to code – [free 3,000-hour curriculum](#)**

entities. This is entirely up to the those creating or designing the database.

Have a look at the diagram below.



ADVERTISEMENT

We have two entities – Customer and Pizza. These entities are linked together by a horizontal line with symbols/indicators/notations.

. . . . .

[Forum](#)[Donate](#)

[Learn to code — free 3,000-hour curriculum](#)

created by none (optional) or many customers.

Similarly, the notation on the right side implies that a customer can order **zero or many** pizzas.

The **cardinality** here is the same for both entities. Zero is the minimum while many is the maximum.

The use cases for other crow's foot notation diagrams are the same as the ones in our examples. It all depends on the logic and what you're designing.

## Conclusion

This tutorial serves as an introduction to understanding entity relationship diagrams and crow's foot notation for database design.

We can use entity relationship diagrams to create a

[Forum](#)[Donate](#)

**Learn to code — [free 3,000-hour curriculum](#)**

database. This makes it easier to understand how each entity associates with another.

Notation makes it easier to understand the relationships between entities. In our case, we used crow's foot notation.

We began by explaining some key terminologies associated with entity relationship diagrams and crow's foot notation like entities, attributes, cardinality, and the meaning of the various crow's foot notation diagrams.

We then saw some examples to help understand the application of the crow's foot notation diagrams in defining the relationship between entities in an entity relationship diagram.

ADVERTISEMENT

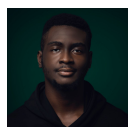
The examples used are very basic, but this isn't always the case when working on an actual database. Having a

[Forum](#)[Donate](#)

Learn to code – [free 3,000-hour curriculum](#)

designs.

Thank you for reading!



**Ihechikara Vincent Abba**

ihechikara.com

---

If you read this far, thank the author to show them you care.

[Say Thanks](#)

Learn to code for free. freeCodeCamp's open source curriculum has helped more than 40,000 people get jobs as developers.

[Get started](#)

ADVERTISEMENT

[Forum](#)[Donate](#)

**Learn to code – [free 3,000-hour curriculum](#)**

freeCodeCamp is a donor-supported tax-exempt 501(c)(3) charity organization (United States Federal Tax Identification Number: 82-0779546)

Our mission: to help people learn to code for free. We accomplish this by creating thousands of videos, articles, and interactive coding lessons - all freely available to the public.

Donations to freeCodeCamp go toward our education initiatives, and help pay for servers, services, and staff.

You can [make a tax-deductible donation here](#).

[Forum](#)

[Donate](#)

## Learn to code – [free 3,000-hour curriculum](#)

[Python Code Examples](#)

[Learn Algorithms in JS](#)

[Learn Swift](#)

[Learn Solidity](#)

[REST API Best Practices](#)

[Command Line for Beginners](#)

[Distributed Systems Patterns](#)

[Open Source for Devs](#)

[How to Write Clean Code](#)

[Learn Golang](#)

[Learn Express.js](#)

[Front-End JS Development](#)

[Intro to Operating Systems](#)

[Software Architecture](#)

[Patterns](#)

[HTTP Networking in JS](#)

[Learn PHP](#)

[Learn Node.js](#)

[Learn JS Modules](#)

[Learn to Build REST APIs](#)

[Learn to Build GraphQL APIs](#)

[Write React Unit Tests](#)

[Learn Java](#)

[Learn CSS Grid](#)

[Learn Apache Kafka](#)

[Intermediate TS and React](#)

[OSS Security Best Practices](#)

## Mobile App



## Our Charity

[About](#) [Alumni Network](#) [Open Source](#) [Shop](#) [Support](#) [Sponsors](#) [Academic Honesty](#) [Code of Conduct](#) [Privacy Policy](#)

[Terms of Service](#) [Copyright Policy](#)