



StinaQ.me

..

- [Blog](#)
- [Archives](#)
- [Portfolio](#)

[GitHub](#)
[RSS](#)

An Easy Way to Finding the Candidate Key of a Database Relation

A beginner's guide to locating the Candidate Key and Normal Form of a relation.

There are many [systems](#) and algorithms that can help you find the candidate key of a particular relation. What these don't tell you is how much easier this can be if you just learn to draw appropriate boxes and arrows! Granted, you might not look like the coolest kid in a group of hardened database programmers if you do this. But we are all beginners in the, you know..beginning.

Swallow your pride, this is a really good and clear way! This article is meant for those of you who are starting out with relational databases and normalization. I hope you know the purpose of normalization and what the different normal forms are.

Every relation has one or more candidate keys. A candidate key (CK) is one or more attribute(s) in a relation with which you can determine all the attributes in the relation. Later on, when you implement the relation in a database, the candidate key, or one of the keys, will be the **primary key** of a relation. Knowing the candidate key is also needed to determine the normal form of the relation. But how do you find it? Well, you can either look all professional and use algorithms. Or you can draw it!

An example of how to draw you way to the candidate key

Let me show you with an example. This relation deals with movies and actors.

Movie(title, year, genre, genre-description, director, director-home-country)

First, we have to list the dependencies. Which attributes are dependent on which? An arrow means that by knowing the one to the left, you know the one to the right.

title \longrightarrow year

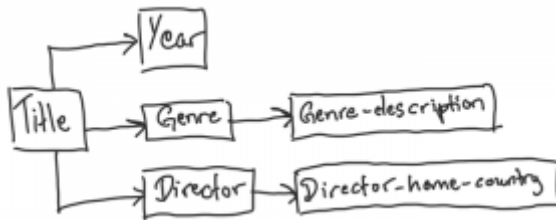
title \longrightarrow genre

title \longrightarrow director

genre \longrightarrow genre-description

director \longrightarrow director-home-country

So what I do is that I make a box for each of the attributes and then simply add arrows to represent dependencies.



Now it's extremely easy to locate the least amount of attributes you need to determine the rest of the relation:

Title! No arrows are pointing at title and all the other boxes have arrows that originate from title.

Therefore: **The candidate key of this relation is Title**, and that makes title a primary attribute (an attribute that is part of candidate key). More on primary attributes later, when we continue on to deciding the normal form.

This example was pretty basic and simple, let's continue on to something a bit trickier. And let's also leave this with attributes that makes sense behind; my imagination was stretched a little too thin by the last example...

Consider the relation:

$R(A, B, C, D, E)$

where

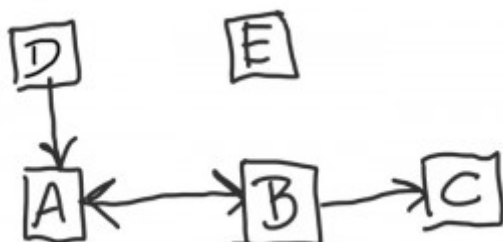
$A \longrightarrow B$

$B \longrightarrow A$

$B \longrightarrow C$

$D \longrightarrow A$

Then you'll get this drawing:



No arrow is pointing at neither D or E, so in order to determine all the attributes in this relation you'll need to know both D and E, since nothing can determine either of them. This gives you a composite key. **Candidate key: {D, E}**. C and E together. I'll show just one more example before moving on to using this knowledge. A trickier and larger one.

$R(A, B, C, D, E, F, G)$

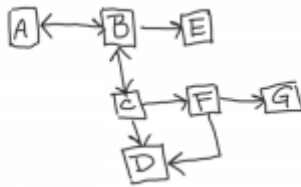
$A \longrightarrow B$

$B \longrightarrow \{A, C, E\}$

$C \longrightarrow \{B, F, D\}$

$F \longrightarrow \{D, G\}$

The curly brackets are just a short hand. Instead of saying $A \longrightarrow B$, and $A \longrightarrow C$, you write $A \longrightarrow \{B, C\}$.



Here you actually have several candidate keys. If you know A you can determine the entire relation, but that is also true for B and C. From B you can get to all the other boxes by just following arrows, the same for C and A! So the result is **several candidate keys, A, B and C**.

Using the drawing to determine the normal form of a relation

The drawings are also a great help on the way to find out which normal form a relation has. All you have to do is know how to draw boxes, arrows and what the rules of the [normal forms say](http://stinaq.me/blog/2012/11/21/an-easy-way-to-finding-the-candidate-key-of-a-database-relation/).

So let's try it with a relation.

$R(A, B, C, D)$

$A \longrightarrow B$

$B \longrightarrow C$

$C \longrightarrow \{B, D\}$

Which gives this drawing:



The Candidate Key of this relation is A, since that's all you need to determine the rest of the relation. That makes A the only primary attribute (an attribute that is part of a candidate key). But what of the normal forms? I will go through them one by one.

- **First normal form.** Are all the attributes atomical? Well, **yes!** None of them are more than one, so to speak.
- **Second normal form.** Is there a non primary attribute that's determined by parts of any candidate key? This implies that we need a composite candidate key (one that consist of two or more attributes, like in the second example above), and since we don't, this can't be true and the **relations is therefore in 2nd normal form.** To have a relation that is not in 2nd normal form, there would have to be an arrow pointing *from* an attribute that is part of a candidate key, *towards* an attribute that is not part of a candidate key.
- **Third normal form.** Does a non primary attribute exists that's determined by another non primary attribute? Yes, in this case there is! There is an arrow from B to C and both of them are non primary attributes, so **this relation is not in 3rd normal form.**

What about 4th and Boyce Codd and all the other normal forms, then? Well, a relation that's in 4th normal form must fulfill the rules regarding the 4th normal form as well as all the others "leading up" to it. So, if it breaks the rule for 3rd, it can't be in 4th.

One last example.

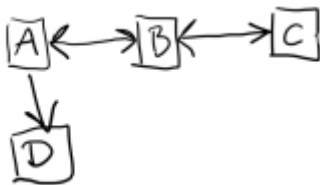
R(A, B, C, D)

$A \longrightarrow \{B, D\}$

$B \longrightarrow \{A, C\}$

$C \longrightarrow B$

The drawing:



And the candidate key will be A, B and C.

- **First normal form.** Are all the attributes atomical? Still yes.
- **Second normal form.** Is there a non primary attribute that's determined by parts of a candidate key? No, the same as above. The only non primary attribute is D and it's not determined by parts of a candidate key.
- **Third normal form.** Does a non primary attribute exists that is determined by another non primary attribute? No, no two non primary attributes determine each other. In this case we only have one, so it can't happen.
- **Boyce-Codd normal form.** Are all the deteminates candidate keys? Or, as I like to think of it, are all attributes to the left of the arrows candidate keys? You just look at where the dependencies are listed (above the drawing) and see if all those to the left are candidate keys. In this case, they are.
- **Fourth normal form.** Deals with the problems that can arise from having multivalued dependencies, and seeing as there are no multivalued dependencies, we can assume that it is good to go for the 4NF!

So, the relation is in 1 to 4th normal form and Boyce-Codd, and we'll leave the rest for another time! I hope this article has made the topic of candidate keys and normal forms somewhat clearer. There are probably many other ways to solve this problem, but this has worked great for me.

Feel very free to contact me with questions or suggestions.

/SQ

