Finding the candidate keys

The first step in the process of finding a normal form and decomposing a relation is to find the candidate keys. This is a set of examples to find them:

Example 1

R = (ABCDE), F = {A -> C, E -> D, B -> C}

Ok , so the first step in finding the candidate keys is to find the attribute closure given F.

A+ = AC
B+ = BC
C+ = C
D+ = D
E+ = DE

From this information we need to find the candidate keys. Any attribute that only appears on the right side in a trivial dependency must be in the candidate key. For this, that includes ABE. Does ABE+ get us to a candidate key? ABE+ = ABCDE – yes it does. The candidate key is ABE.

Example 2

R = ABCDE, F = {A -> BE, C -> BE, B -> D}

Ok, lets compute the attribute closure:

A+ = ABDE
B+ = BD
C+ = CBDE
D+ = D
E+ = E

The 2 attributes that only appear on the right side in a trivial are AC. Is AC a candidate key? Yes.

Example 3

R = ABCDEF, F = {A -> B, B -> D, C -> D, E -> F}

Let's compute the attribute closures:

A+ = ABD
B+ = BD
C+ = CD
D+ = D
E+ = EF

Ok, let's start with those attributes that only appear on the right side in trivial FDs. They are ACE . ACE+ = ABCDEF, so ACE is a candidate key.

Example 4

R = ABCD, F={AB -> C, BC -> D, CD -> A}

Computing the attribute closure:

The single letters are all trivial.

AB+ = ABCD, BC+=ABCD, CD+=ACD

So our candidate keys are AB and BC.. Why isn't BCD+ a candidate key? Because it is not minimal. The D is extraneous since BC -> D.

Example 5

R = ABCD, F={A -> BCD, C -> A}

Attribute closure:

A+ - ABCD
B+ = B
C+ = ABCD
D+ = D

Our candidate keys are A and C.