Steps to find out the Candidate Key of a Relational Table using Functional Dependencies

Step 1:

First, we will find out the essential and non-essential sets of attributes from the given set of attributes. Those attributes which are dependent on other attributes are non-essential attributes, and their values can be found out using essential attributes.

So, all the essential attributes will definitely be part of our candidate key.

Step 2:

We will combine all the essential attributes, and if they can determine all the attributes (by finding their closure), then it will be the candidate key.

Step 3:

If a combination of essential attributes is not a candidate key, then we will try different combinations of essential and non-essential attributes which are not subset of each other to find out all the candidate keys.

Example 1:

Let's suppose we have a set of attributes as S: {A, B, C, D} and functional dependencies are:

A -> B

B -> C

C -> A

Solution:

In the above example, we have non-essential attributes as {A, B, C}, and the essential attribute is {D}.

So, D will be the part of the candidate key.

Closure of D: {D}

So, a combination of essential attributes is not able to find all the attributes, so we will add non-essential attributes in different ways to find out the candidate keys.

Closure of (AD) = {A, B, C, D} (using A->B and B->C)

So, AD will be a candidate key.

Closure of (BD) = {A, B, C, D} (using C->A and B->C)

So, BD will be a candidate key.

Closure of (CD) = {A, B, C, D} (using C->A and A->B)

So, CD will be a candidate key.

No other combination of attributes is not possible, so candidate keys are {AD, BD, CD}

Example 2:

Let's suppose we have a set of attributes as S: {A, B, C, D, E, F} and functional dependencies are:

AB -> C

B -> AE

C -> D

Solution:

In the above example, we have non-essential attributes as {A, C, D, E}, and the essential attribute is {B, F}.

So, BF will be the part of the candidate key.

Closure of BF: {B, A, E, C, D, F}

Since the combination of essential attributes is capable of finding all the attributes, it will be a candidate key. All other combinations of the essential and non-essential attributes will not be minimal, so there will be only one candidate key as {BF}.

Example 3:

Let's suppose we have a set of attributes as S: {A, B, C, D, E} and functional dependencies are:

CE -> D

D -> B

C -> A

Solution:

In the above example, we have non-essential attributes as {A, B, D}, and the essential attribute is {C, E}.

So, CE will be the part of the candidate key.

Closure of CE: {C,E,D,B,A} using (CE->D, D->B and C->A)

Since the combination of essential attributes is capable of finding all the attributes, it will be a candidate key. All other combinations of the essential and non-essential attributes will not be minimal, so there will be only one candidate key as {CE}.

Example 4:

Let's suppose we have a set of attributes as S: {W, X, Y, Z} and functional dependencies are:

Z -> W

Y -> XZ

XW -> Y

Solution:

In the above example, we have non-essential attributes as {W, X, Y, Z}, and the essential attribute is {}.

In this example, we will explore each and every combination of attributes to find out the candidate key, as there is no essential attribute.

Closure of W = {W}

Closure of X = {X}

Closure of Y = {Y,X,Z,W} (using Y->XZ and Z->W)

Closure of Z = {ZW}

Since Y is capable of finding all the attributes so it will be a candidate key. Y will not be part of the next combinations.

Closure of XW = {X,W,Y,Z} (using XW->Y and Y->XZ)

Closure of ZW = {Z,W}

Closure of XZ = {X,W,Y,Z} (using Z->W and XW->Y)

So XW and XZ will be the next candidate keys, and the next other combinations will contain no attributes apart from these candidate keys.

So there will be three candidate keys which are {Y, XW, XZ}

Example 5:

Let's suppose we have a set of attributes as S: {A, B, C, D, E, F} and functional dependencies are:

AB -> C

C-> D

D -> BE

E -> F

F -> A

Solution:

In the above example, we have non-essential attributes as {A, B, C, D, E, F}, and the essential attribute is {}.

Now in this example, we will explore each and every combination of attributes to find out the candidate key, as there is no essential attribute.

If there is no essential attribute, then each non-essential attribute can be a candidate key, so it increases the number of candidate keys.

Closure of A = {A}

Closure of B = {B}

Closure of C = {C, D, B, E, F, A} using (C->D, D->BE, E->F, F->A)

Closure of D = {D, B, E, F, A, C} using (AB->C, D->BE, E->F, F->A)

Closure of E = {E,F,A} using (A->F, F->A)

Closure of F = {F,A} using(F->A)

Since {C, D} are capable of finding all the attributes so it will be a set of candidate keys. {C, D} will not be part of the next combinations.

Closure of AB = { A, B, C, D, E, F}

Closure of AE = { A,E,F}

Closure of AF = { A,F}

Closure of BE = { A, B, C, D, E, F}

Closure of BF = { A, B, C, D, E, F}

Closure of EF = { A,E,F}

So AB, BE, and BF will be the next candidate keys, and the next other combinations will contain no attributes apart from these candidate keys.

So there will be five candidate keys which are {C, D, AB, BE, BF}