LET'S START WITH DBMS :)

Attribute closure

5. Now lets find candidate key, super key , prime and non-prime attributes

Super key : Set of attributes whose closure contains all the attributes given in a relation Super set of any candidate key is super key. A key(combination of all possible attributes) which can uniquely identify two tuples.

How to find Super Key?

- Identify All Attributes
- 2. Analyze the functional dependencies and find closure.
- Generate Power Set : If A has n attributes, the power set will have 2^n subsets.
- Check for Super Key Property : For each subset in the power set, check if it can uniquely identify each tuple in the relation.

LET'S START WITH DBMS :)

Attribute closure

Q. Find all the superkeys for the relation R(A, B, C) and FD : A → B, B → C.

1.Identify All Attributes:

A={A,B,C}

2. Analyze the functional dependencies and find closure.

From A → B and B → C, we can infer A → C

A+={A,B,C}

LET'S START WITH DBMS :)

Attribute closure

B+ = {B,C} because B → C, but B does not determine A.

C+ ={C}

Clsoure of A gives or determins all the attributes in the table so we can say its s auper key.

3. How to find all the super keys?

Find the power subset

The power set of {A,B,C} is 2^3=8

{}, {A}, {B}, {C}, {A, B}, {A, C}, {B, C}, {A, B, C}

LET'S START WITH DBMS :)

Attribute closure

A power set is the set of all subsets of a given set, including the empty set and the set itself. If you have a set X, the power set of X is denoted as 2^X

Example:

Let's take a simple set X={a,b}
The power set of X would include the following subsets:

1. The empty set: ∅
2. The single-element subsets: {a} & {b}
3. The full set itself: {a,b}

So, the power set P(X) would be: P(X)={∅,{a},{b},{a,b}}

LET'S START WITH DBMS :)

Attribute closure

4. Verify each subset to see if it is a super key

1. {}: Not a super key.
2. {A}: Super key (as its closure determines all the attributes in relation).
3. {B}: Not a super key (does not determine A).
4. {C}: Not a super key (does not determine A or B).
5. {A, B}: Super key (A is already a super key, so adding B still keeps it a super key).
6. {A, C}: Super key (A is already a super key, so adding C still keeps it a super key).
7. {B, C}: Not a super key (B determines C, but does not determine A).
8. {A, B, C}: Super key (A is already a super key, so adding B and C keeps it a super key).

LET'S START WITH DBMS :)

Attribute closure

5. Super keys are : {A}, {A, B}, {A, C}, {A, B, C}

We can also say to find max number of super keys we can use the formula

2^n-k

where

k- candidate key with k attributes (k < n) in a relation

n- total no of attributes in a relation

When 1 C.K is there = 2^n-1

When 2 C.K is there = 2^n-1(for 1st ck)+2^n-1(for 2nd ck)-2^n-2(for combination of both)