LET'S START WITH DBMS :)

Attribute closure

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

Candidate key : A superkey whose proper subset is not a super key

A set A is considered a proper subset of set B

- All elements of A are also elements of B
- Set B has at least one element that is NOT in A.

Prime Attribute: An attribute that is part of any candidate key.
Non-Prime Attribute: An attribute that is not part of any candidate key.

LET'S START WITH DBMS :)

Attribute closure

How to find the number of candidate keys?

1. Take the closure of the entire attribute set
2. Discard the dependents if their deteminants are present
3. After discarding the final combination you get is a candidate key if its subset doesnt have a super key and mention them in prime attribute
4. Now check all the functional dependencies and see if in the right hand side is there a attribute which has a determinant present in the prime attribute, if yes mention that as well in the prime attribute and replace the dependent attribute with detrminant and check if its candidate key.
5. If no prime attribute is available in teh right hand side of the functional dependency we can say there are no more candidate key and the one key we made after discarding dependent and checking for super key, that is the only candidate key.

LET'S START WITH DBMS :)

Attribute closure

Q. Find no of candidate and super key for the given relation R (A,B,C,D) and functional dependency A->B, B->C, C->D , B->A

1. Lets find the minimal superkeys
2. We will get the value of k and n ,where k- candidate key with k attributes (k < n) in a relation , n- total no of attributes in a relation
3. Use this to get the super keys in the relation 2^n-k

LET'S START WITH DBMS :)

How to find the number of candidate keys?

1. Take the closure of the entire attribute set
2. Discard the dependents if their deteminants are present
3. After discarding the final combination you get is a candidate key if its subset doesnt have a super key and mention them in prime attribute
4. Now check all the functional dependencies and see if in the right hand side is there a attribute which has a determinant present in the prime attribute, if yes mention that as well in the prime attribute and replace the dependent attribute with detrminant and check if its candidate key.
5. If no prime attribute is available in teh right hand side of the functional dependency we can say there are no more candidate key and the one key we made after discarding dependent and checking for super key, that is the only candidate key.

LET'S START WITH DBMS :)

How to find the number of candidate keys?

1. Take the closure of the entire attribute set
2. Discard the dependents if their deteminants are present
3. After discarding the final combination you get is a candidate key if its subset doesnt have a super key and mention them in prime attribute
4. Now check all the functional dependencies and see if in the right hand side is there a attribute which has a determinant present in the prime attribute, if yes mention that as well in the prime attribute and replace the dependent attribute with detrminant and check if its candidate key.
5. If no prime attribute is available in teh right hand side of the functional dependency we can say there are no more candidate key and the one key we made after discarding dependent and checking for super key, that is the only candidate key.