CSE 301 – DATABASE

Lecture 11

Chapter 5: Relational Database Design

Minimal Cover or Canonical Cover or Irreducible set of Functional Dependencies

- A canonical cover or irreducible a set of functional dependencies FD is a simplified set of FD that has a similar closure as the original set FD.

- The formal definition is A set of FD F to be minimal if it satisfies the following conditions:
  - ✓ Every dependency in F has a single attribute for its right-hand side.
  - ✓ Do not replace any dependency X → A in F with a dependency Y→ A, where Y is a proper subset of X, and still have a set of dependencies that is equivalent to F.
  - ✓ We cannot remove any dependency from F and still have a set of dependencies that are equivalent to F.

Minimal Cover or Canonical Cover or Irreducible set of FD

- A canonical cover is a simplified and reduced version of the given set of functional dependencies.
- Since it is a reduced version, it is also called as Irreducible set.
- Canonical cover is free from all the extraneous functional dependencies.
- The closure of canonical cover is same as that of the given set of functional dependencies.
- Canonical cover is not unique and may be more than one for a given set of functional dependencies.

Minimal Cover or Canonical Cover or Irreducible set of FD

Why we need Minimal Cover or Canonical Cover ?

- ✓ Working with the set containing extraneous functional dependencies increases the computation time.
- ✓ Therefore, the given set is reduced by eliminating the useless functional dependencies.
- ✓ This reduces the computation time and working with the irreducible set becomes easier.

Steps to Find Canonical Cover

- Step-01: Write the given set of functional dependencies in such a way that each functional dependency contains exactly one attribute on its right side.

- Example:

The functional dependency $X \rightarrow YZ$ will be written as

$X \rightarrow Y$

$X \rightarrow Z$

Steps to Find Canonical Cover

- Step-02: Consider each functional dependency one by one from the set obtained in Step-01 and determine whether it is essential or non-essential.
- To determine whether a functional dependency is essential or not, compute the closure of its left side-
  - ✓ Once by considering that the particular functional dependency is present in the set
  - ✓ Once by considering that the particular functional dependency is not present in the set.
  - ✓ Then following two cases are possible:

Steps to Find Canonical Cover

- Case-01: Results Come Out to be Same
  - ✓ It means that the presence or absence of that functional dependency does not create any difference.
  - ✓ Thus, it is non-essential.
  - ✓ Eliminate that functional dependency from the set.
  - ✓ Eliminate the non-essential functional dependency from the set as soon as it is discovered.
  - ✓ Do not consider it while checking the essentiality of other functional dependencies.

Steps to Find Canonical Cover

- Case-02: Results Come Out to be Different
  - ✓ It means that the presence or absence of that functional dependency creates a difference.
  - ✓ Thus, it is essential.
  - ✓ Do not eliminate that functional dependency from the set.
  - ✓ Mark that functional dependency as essential.

Steps to Find Canonical Cover

- Step-03:
  - ✓ Consider the newly obtained set of functional dependencies after performing Step-02.
  - ✓ Check if there is any functional dependency that contains more than one attribute on its left side.
  - ✓ Then following two cases are possible:

Steps to Find Canonical Cover

- Case-01: No
  - ✓ There exists no functional dependency containing more than one attribute on its left side.
  - ✓ In this case, the set obtained in Step-02 is the canonical cover.
- Case-02: Yes
  - ✓ There exists at least one functional dependency containing more than one attribute on its left side.
  - ✓ In this case, consider all such functional dependencies one by one.
  - ✓ Check if their left side can be reduced.

Steps to Find Canonical Cover

- Case-02: Yes : Use the following steps to perform a check:
  - ✓ Consider a functional dependency.
  - ✓ Compute the closure of all the possible subsets of the left side of that functional dependency.
  - ✓ If any of the subsets produce the same closure result as produced by the entire left side, then replace the left side with that subset.
  - ✓ After this step is complete, the set obtained is the canonical cover.

Example of Canonical Cover

- The following functional dependencies hold true for the relational scheme

  R (W , X , Y , Z)

  $X \rightarrow W$

  $WZ \rightarrow XY$

  $Y \rightarrow WXZ$

Find the Minimal Cover or Canonical Cover or irreducible set of FD's equivalent for this set of functional dependencies?

Example of Canonical Cover

- **Step-01:** Write all the functional dependencies such that each contains exactly one attribute on its right side. Here we are using decomposition rule.

  $X \rightarrow W$

  $WZ \rightarrow X$

  $WZ \rightarrow Y$

  $Y \rightarrow W$

  $Y \rightarrow X$

  $Y \rightarrow Z$

Example of Canonical Cover

- Step-02: Check the essentiality of each functional dependency one by one.

- For $X \rightarrow W$:
  - ✓ Considering $X \rightarrow W$, $(X)^+ = \{ X , W \}$
  - ✓ Ignoring $X \rightarrow W$, $(X)^+ = \{ X \}$
- Now,
  - ✓ Clearly, the two results are different.
  - ✓ Thus, we conclude that $X \rightarrow W$ is essential and can not be eliminated

- For $WZ \rightarrow X$:
  - ✓ Considering $WZ \rightarrow X$, $(WZ)^+ = \{ W , X , Y , Z \}$
  - ✓ Ignoring $WZ \rightarrow X$, $(WZ)^+ = \{ W , X , Y , Z \}$
- Now,
  - ✓ Clearly, the two results are same.
  - ✓ Thus, we conclude that $WZ \rightarrow X$ is non-essential and can be eliminated.

Example of Canonical Cover

- Eliminating $WZ \rightarrow X$, our set of functional dependencies reduces to

  $X \rightarrow W$

  $WZ \rightarrow Y$

  $Y \rightarrow W$

  $Y \rightarrow X$

  $Y \rightarrow Z$

- Now, we will consider this reduced set in further checks.

Example of Canonical Cover

- Step-02: Check the essentiality of each functional dependency one by one.

- For $WZ \rightarrow Y$:
  - ✓ Considering $WZ \rightarrow Y$, $(WZ)+ = \{ W , X , Y , Z \}$
  - ✓ Ignoring $WZ \rightarrow Y$, $(WZ)+ = \{ W , Z \}$
- Now,
  - ✓ Clearly, the two results are different.
  - ✓ Thus, we conclude that $WZ \rightarrow Y$ is essential and can not be eliminated

- For $Y \rightarrow W$:
  - ✓ Considering $Y \rightarrow W$, $(Y)+ = \{ W , X , Y , Z \}$
  - ✓ Ignoring $Y \rightarrow W$, $(Y)+ = \{ W , X , Y , Z \}$
- Now,
  - ✓ Clearly, the two results are same.
  - ✓ Thus, we conclude that $Y \rightarrow W$ is non-essential and can be eliminated.

Example of Canonical Cover

- Eliminating $Y \rightarrow W$, our set of functional dependencies reduces to

  $X \rightarrow W$

  $WZ \rightarrow Y$

  $Y \rightarrow X$

  $Y \rightarrow Z$

- Now, we will consider this reduced set in further checks.

Example of Canonical Cover

- Step-02: Check the essentiality of each functional dependency one by one.

- For $Y \rightarrow X$:
  - ✓ Considering $Y \rightarrow X$, $(Y)+ = \{ W , X , Y , Z \}$
  - ✓ Ignoring $Y \rightarrow X$, $(Y)+ = \{ Y , Z \}$
- Now,
  - ✓ Clearly, the two results are different.
  - ✓ Thus, we conclude that $Y \rightarrow X$ is essential and can not be eliminated

- For $Y \rightarrow Z$:
  - ✓ Considering $Y \rightarrow Z$, $(Y)+ = \{ W , X , Y , Z \}$
  - ✓ Ignoring $Y \rightarrow Z$, $(Y)+ = \{ W , X , Y \}$
- Now,
  - ✓ Clearly, the two results are different.
  - ✓ Thus, we conclude that $Y \rightarrow Z$ is essential and can not be eliminated.

Example of Canonical Cover

- From here, our essential functional dependencies are

  $X \rightarrow W$

  $WZ \rightarrow Y$

  $Y \rightarrow X$

  $Y \rightarrow Z$

Example of Canonical Cover

- Step-03: Consider the functional dependencies having more than one attribute on their left side.
  - ✓ Check if their left side can be reduced.
  - ✓ In our set, Only $WZ \rightarrow Y$ contains more than one attribute on its left side.
  - ✓ Considering $WZ \rightarrow Y$,

    $(WZ)^+ = \{ W , X , Y , Z \}$
  - ✓ Now, Consider all the possible subsets of WZ.
  - ✓ Check if the closure result of any subset matches to the closure result of WZ.

    $(W)^+ = \{ W \}$

    $(Z)^+ = \{ Z \}$

Example of Canonical Cover

- Clearly, None of the subsets have the same closure result same as that of the entire left side. Thus, we conclude that we can not write $WZ \rightarrow Y$ as $W \rightarrow Y$ or $Z \rightarrow Y$. Thus, set of functional dependencies obtained in step-02 is the canonical cover.

- Finally, the canonical cover is

$X \rightarrow W$		$X \rightarrow W$
$WZ \rightarrow Y$		$WZ \rightarrow Y$
$Y \rightarrow X$		$Y \rightarrow XZ$
$Y \rightarrow Z$		

Exercise on Minimal Cover or Canonical Cover

- R (ABCD)

FD: $\{A \rightarrow B, C \rightarrow B, D \rightarrow ABC, AC \rightarrow D\}$

- Find the minimal cover.

- R (VWXYZ)
- FD: $\{V \rightarrow W, VW \rightarrow X, Y \rightarrow VXZ\}$
- Find the canonical cover.

- R (ABC)
- FD: $\{A \rightarrow BC, B \rightarrow AC, C \rightarrow AB\}$

- R (ABCD)

FD: $\{A \rightarrow BC, B \rightarrow C, A \rightarrow B, AB \rightarrow C\}$

- Find the irreducible set of FD

- R (ABCDE)
- FD: $\{A \rightarrow BC, CD \rightarrow E, B \rightarrow D, E \rightarrow A\}$
- Find the canonical cover.

- R (ABC)
- FD: $\{A \rightarrow B, B \rightarrow C, C \rightarrow A\}$
- Find the minimal cover.

Exercise on Minimal Cover or Canonical Cover

- R (ABC)

FD: $\{A \to C, AB \to C\}$

- R (ABC)

FD: $\{A \to BC, B \to C, A \to B, AB \to C\}$

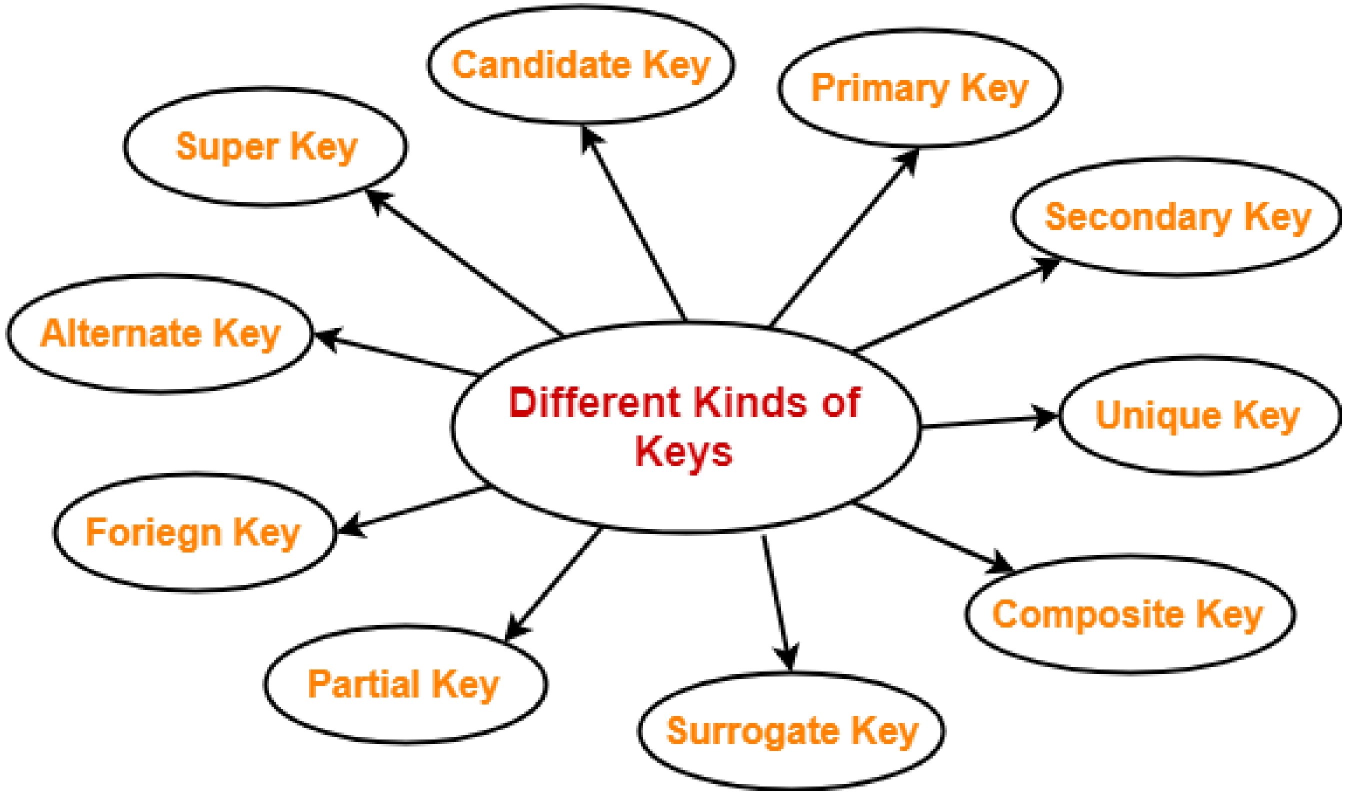
- R (ABCD)
- F: $\{ABC \to CD, BC \to D, A \to B, C \to D\}$

- R (ABCDE)
- F: $\{ABCD \to E, E \to D, A \to B, AC \to D\}$

- R (ABCDEH)
- F: $\{A \to BC, CD \to E, E \to C, D \to EAH, ABH \to BD, DH \to BC\}$

Keys in DBMS

- A key is a set of attributes that can identify each tuple uniquely in the given relation.
- There are following 10 important keys in DBMS



Super Key

- A super key is a attribute or a set of attributes that can identify each tuple uniquely in the given relation.
- A super key is not restricted to have any specific number of attributes.
- Thus, a super key may consist of any number of attributes.

Super Key

- Example: Consider the following Student schema

  Student (roll , name , sex , age , address , class , section)

- Given below are the examples of super keys since each set can uniquely identify each student in the Student table

  (roll , name , sex , age , address , class , section),

  (class , section , roll),

  (class , section , roll , sex),

  (name , address),

  etc.

- NOTE: All the attributes in a super key are definitely sufficient to identify each tuple uniquely in the given relation but all of them may not be necessary.

Candidate Key

- A minimal super key is called as a candidate key.
- Example- Consider the following Student schema

  Student (roll , name , sex , age , address , class , section)

- Given below are the examples of candidate keys since each set consists of minimal attributes required to identify each student uniquely in the Student table

  (class , section , roll)

  (name , address)

Candidate Key

- NOTES
  - ✓ All the attributes in a candidate key are sufficient as well as necessary to identify each tuple uniquely.
  - ✓ Removing any attribute from the candidate key fails in identifying each tuple uniquely.
  - ✓ The value of candidate key must always be unique.
  - ✓ The value of candidate key can never be NULL.
  - ✓ It is possible to have multiple candidate keys in a relation.
  - ✓ Those attributes which appears in some candidate key are called as prime attributes.

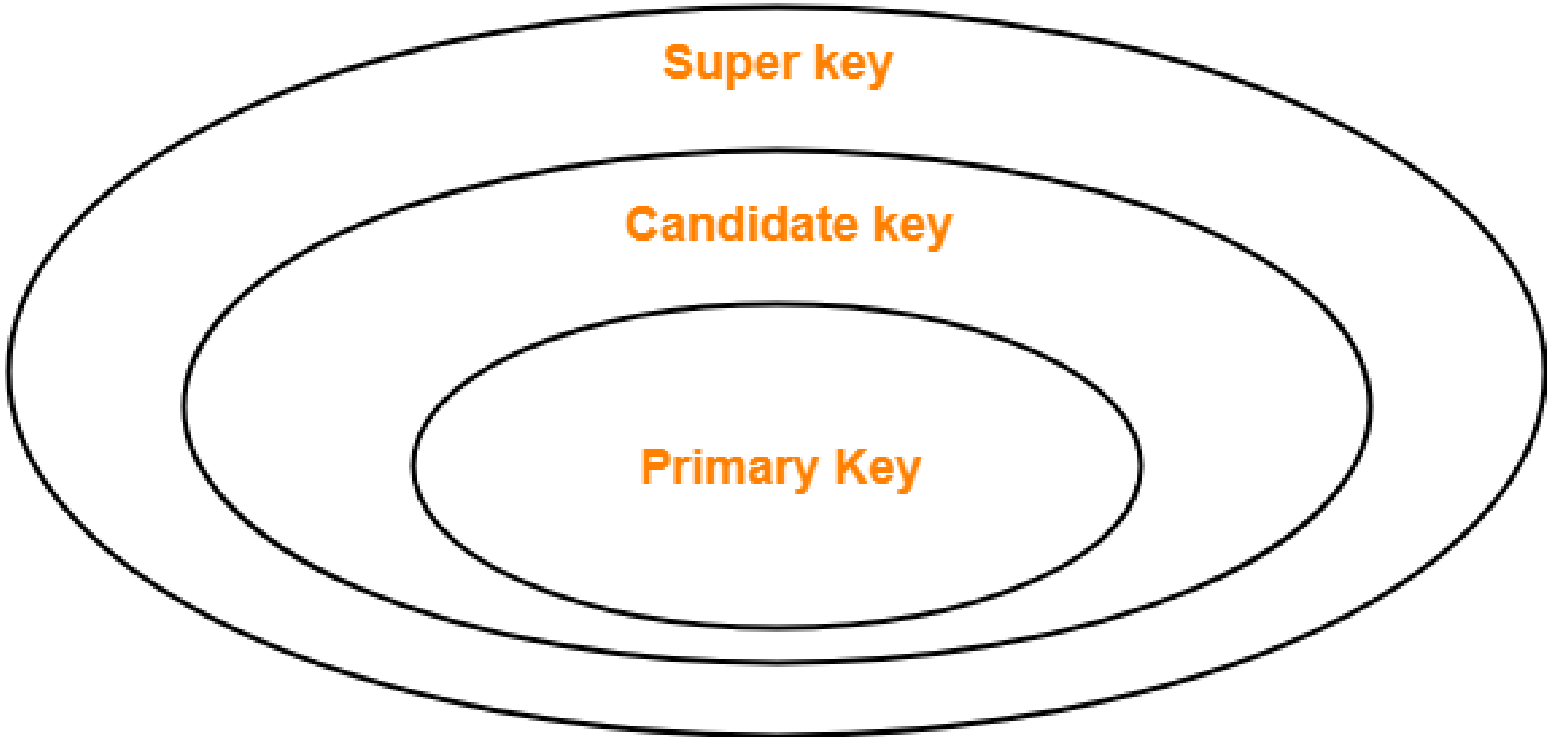
Primary Key

- A primary key is a candidate key that the database designer selects while designing the database.

OR

- Candidate key that the database designer implements is called as a primary key.
- NOTES:
  - ✓ The value of primary key can never be NULL.
  - ✓ The value of primary key must always be unique.
  - ✓ The values of primary key can never be changed i.e. no updation is possible.
  - ✓ The value of primary key must be assigned when inserting a record.
  - ✓ A relation is allowed to have only one primary key.

Primary Key / Super Key / Primary Key



Alternate Key

- Candidate keys that are left unimplemented or unused after implementing the primary key are called as alternate keys.

OR

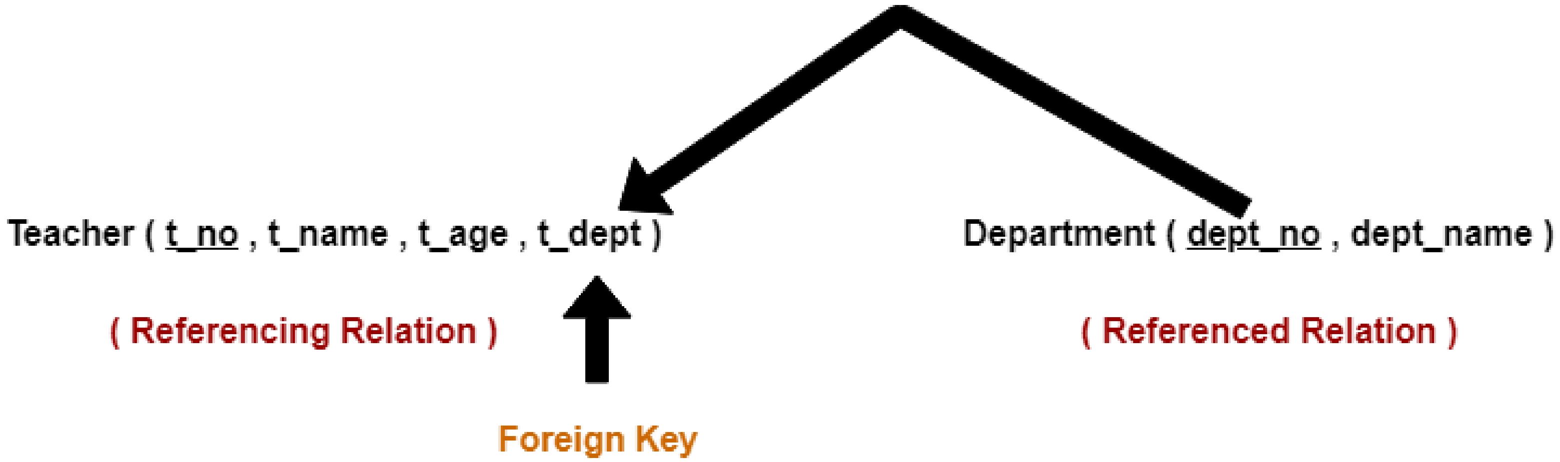
- Unimplemented candidate keys are called as alternate keys.

Foreign Key

- An attribute ‘X’ is called as a foreign key to some other attribute ‘Y’ when its values are dependent on the values of attribute ‘Y’.
- The attribute ‘X’ can assume only those values which are assumed by the attribute ‘Y’.
- Here, the relation in which attribute ‘Y’ is present is called as the referenced relation.
- The relation in which attribute ‘X’ is present is called as the referencing relation.
- The attribute ‘Y’ might be present in the same table or in some other table.

Foreign Key

- Consider the following two schemas



- Here, t_dept can take only those values which are present in dept_no in Department table since only those departments actually exist.

Foreign Key

- NOTES:
  - Foreign key references the primary key of the table.
  - Foreign key can take only those values which are present in the primary key of the referenced relation.
  - Foreign key may have a name other than that of a primary key.
  - Foreign key can take the NULL value.
  - There is no restriction on a foreign key to be unique.
  - In fact, foreign key is not unique most of the time.
  - Referenced relation may also be called as the master table or primary table.
  - Referencing relation may also be called as the foreign table.

Partial Key

- Partial key is a key using which all the records of the table can not be identified uniquely.
- a bunch of related tuples can be selected from the table using the partial key.
- Consider the following schema

    Department (Emp_no , Dependent_name , Relation)

- Here, using partial key Emp_no, we can not identify a tuple uniquely but we can select a bunch of tuples from the table.

Emp_no	Dependent_name	Relation
E1	Suman	Mother
E1	Ajay	Father
E2	Vijay	Father
E2	Ankush	Son

Composite Key

- A primary key comprising of multiple attributes and not just a single attribute is called as a composite key.
- For example, R(ABCDE) is a relation where AC together is a primary key. Then AB is a composite key.

Unique Key

- Unique key is a key with the following properties:
  - ✓ It is unique for all the records of the table.
  - ✓ Once assigned, its value can not be changed i.e. it is non-updatable.
  - ✓ It may have a NULL value.
- Example: The best example of unique key is Social Security Number (SSN).
  - ✓ The Social Security Number is unique for all the citizens (tuples) of a country (table). If it gets lost and another duplicate copy is issued, then the duplicate copy always has the same number as before. Thus, it is non-updatable.
  - ✓ Few citizens may not have got their SSN, so for them its value is NULL.

Surrogate Key

- Surrogate key is a key with the following properties-
  - ✓ It is unique for all the records of the table.
  - ✓ It is updatable.
  - ✓ It can not be NULL i.e. it must have some value.
- Example
  - Mobile Number of students in a class where every student owns a mobile phone.

Secondary Key

- Secondary key is required for the indexing purpose for better and faster searching.

Finding Candidate Keys

- We can determine the candidate keys of a given relation using the following steps:
- Step 01:
  - ✓ Determine all essential attributes of the given relation.
  - ✓ Essential attributes are those attributes which are not present on RHS of any functional dependency.
  - ✓ Essential attributes are always a part of every candidate key.
  - ✓ This is because they can not be determined by other attributes.

Finding Candidate Keys

- **Step 01 Example:** Let R(A, B, C, D, E, F) be a relation scheme with the following functional dependencies

  $A \rightarrow B$

  $C \rightarrow D$

  $D \rightarrow E$

  - ✓ Here, the attributes which are not present on RHS of any functional dependency are A, C and F.
  - ✓ So, essential attributes are: A, C and F.

Finding Candidate Keys

- Step 02:
  - ✓ The remaining attributes of the relation are non-essential attributes.
  - ✓ This is because they can be determined by using essential attributes.
  - ✓ Now, following two cases are possible
  - ✓ Case-01: If all essential attributes together can determine all remaining non-essential attributes, then
    - ❖ The combination of essential attributes is the candidate key.
    - ❖ It is the only possible candidate key.

Finding Candidate Keys

- Step 02:
  - ✓ Case-02: If all essential attributes together can not determine all remaining non-essential attributes, then
  - ✓ The set of essential attributes and some non-essential attributes will be the candidate key(s).
  - ✓ In this case, multiple candidate keys are possible.
  - ✓ To find the candidate keys, we check different combinations of essential and non-essential attributes.

Example of Finding Candidate Keys

- Let $R = (A, B, C, D, E, F)$ be a relation scheme with the following dependencies:

  FD: $\{C \rightarrow F, E \rightarrow A, EC \rightarrow D, A \rightarrow B\}$

  1. Which of the following is a key for R?

     a) CD b) EC c) AE d) AC

  2. Find the total number of candidate key and super keys is possible?

Finding Candidate Keys

- Step 01:
  - ✓ Determine all essential attributes of the given relation.
    - ❖ Essential attributes are those attributes which are not present on RHS of any functional dependency.
  - ✓ So, essential attributes of the relation R are C and E.
  - ✓ So, attributes C and E will definitely be a part of every candidate key.

Finding Candidate Keys

- Step 02: We will check if the essential attributes together can determine all remaining non-essential attributes.

  - ✓ To check, we find the closure of CE.

So, $\{ CE \}^+ = \{ C , E \} = \{ C , E , F \}$ (Using $C \rightarrow F$)

$= \{ A , C , E , F \}$ (Using $E \rightarrow A$)

$= \{ A , C , D , E , F \}$ (Using $EC \rightarrow D$)

$= \{ A , B , C , D , E , F \}$ (Using $A \rightarrow B$)

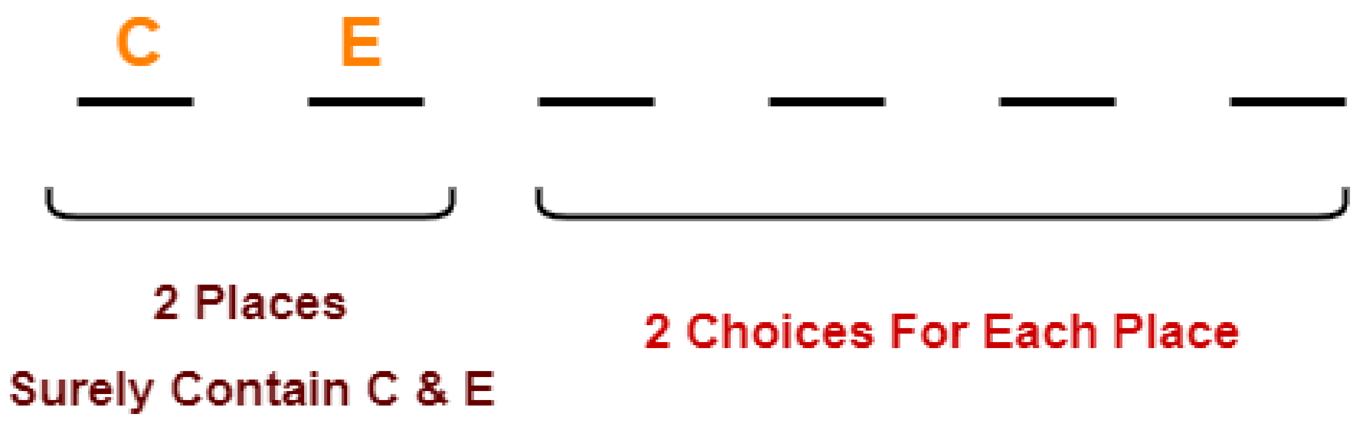
- ❖ We conclude that CE can determine all the attributes of the given relation. So, CE is the only possible candidate key of the relation. Thus, Option (B) is correct.

Finding Total number of Candidate and Super Keys

- **Total Number of Candidate Keys:**
  - ✓ Only one candidate key CE is possible.
- **Total Number of Super Keys:**
  - ✓ There are total 6 attributes in the given relation of which
  - ✓ There are 2 essential attributes- C and E.
  - ✓ Remaining 4 attributes are non-essential attributes.
  - ✓ Essential attributes will be definitely present in every key.
  - ✓ Non-essential attributes may or may not be taken in every super key.

Finding Super Keys

- Total Number of Super Keys:
  - ✓ Thus, total number of super keys possible = 12.



Example 2: Finding Candidate Key

- Let R = (A, B, C, D) be a relation scheme with the following dependencies-

  FD: $\{A \rightarrow B, B \rightarrow C, C \rightarrow A\}$

  Determine the total number of candidate keys and super keys.

- Solution:

  We will find candidate keys of the given relation in the following steps:

- Step-01:
  - ✓ Determine all essential attributes of the given relation.
  - ✓ Essential attributes of the relation is D.
  - ✓ So, attribute will definitely be a part of every candidate key.

Example 2: Finding Candidate Key

- Step-02: $R = (A, B, C, D)$ FD: $\{A \rightarrow B, B \rightarrow C, C \rightarrow A\}$
  - ✓ So, $\{ D \}^+ = \{ D \}$
  - ✓ We can not find R from D^+ , So it D is not a candidate key. It will be the part of candidate key.
  - ✓ Multiple candidate key possible in this relation.
  - ✓ The set of essential attributes and some non-essential attributes will be the candidate key(s). Combinations of essential and non-essential attributes are:
  - ✓ {A, D}, {B, D}, {C, D}, {A, B, D}, {B, C, D}, {A, C, D}
  - ✓ Now find the closure of them and check they are candidate key or not?

Example 2: Finding Candidate Key

- R = (A, B, C, D)   FD: {$A \rightarrow B$, $B \rightarrow C$, $C \rightarrow A$}
  - ✓ So, $\{A, D\}^+ = \{A, D\} = \{A, D, B\}$ (Using $A \rightarrow B$)
  - $= \{A, D, B, C\}$ (Using $B \rightarrow C$) = R   So, {A, D} is a candidate key.
  - ✓ So, $\{B, D\}^+ = \{B, D\} = = \{B, D, C\}$ (Using $B \rightarrow C$)
  - $= \{A, D, B, C\}$ (Using $C \rightarrow A$) = R   So, {B, D} is a candidate key.
  - ✓ So, $\{C, D\}^+ = \{C, D\} = = \{A, C, D\}$ (Using $C \rightarrow A$)
  - $= \{A, B, C, D\}$ (Using $C \rightarrow A$) = R   So, {C, D} is a candidate key.
  - ✓ {A, D}, {B, D}, {C, D} are candidate keys.
  - ✓ Total number of candidate key is 3.

Example 2: Finding Super Key

- R = (A, B, C, D) FD: {A → B, B → C, C → A}
  - ✓ {A, D}, {B, D}, {C, D} are candidate keys.
  - ✓ Combining any attributes with candidate key becomes super key.
  - ✓ So, {A, B, D}, {B, C, D}, {A, C, D} will be the super key because these are super set of {A, D}, {B, D}, {C, D}.
  - ✓ Possible super keys are: {A, D}, {B, D}, {C, D}, {A, B, D}, {B, C, D}, {A, C, D}, {A, B, C, D}.
  - ✓ Total number of super key is 7.

Example 3: Finding Candidate Key

- Let $R = (A, B, C, D)$ be a relation scheme with the following dependencies-

  FD: $\{AB \rightarrow CD, C \rightarrow A, D \rightarrow B\}$

  Determine the total number of candidate keys and super keys.

- Solution:

We will find candidate keys of the given relation in the following steps:

- Step-01:

  - ✓ Determine all essential attributes of the given relation.
  - ✓ Here, no essential attribute (all the attributes are present in RHS of the relation).
  - ✓ So, attribute will definitely be a part of every candidate key.

Example 3: Finding Candidate Key

- Let R = (A, B, C, D)   FD: {AB $\rightarrow$ CD, C $\rightarrow$ A, D $\rightarrow$ B}
  - ✓ Essential attributes of the relation is D.
  - ✓ So, find the closure of individuals and combinations of them and check they are candidate key or not?
  - ✓ First, check individuals:

    {A}+ = {A} = {A}   So, No C. K.

    {B}+ = {B} = {B}

    {C}+ = {C} = {C, A} (Using C $\rightarrow$ A) So, No C. K.

    {D}+ = {D} = {D, B} (Using D $\rightarrow$ B) So, No C. K.

Example 3: Finding Candidate Key

- Let R = (A, B, C, D) FD: {AB → CD, C → A, D → B}
  - ✓ Second, combination of A, B, C, D:
  - ✓ {A, B}+ = {A, B} = {A, B, C, D} (Using AB → CD) So, {A, B} is a C. K.
  - ✓ {A, C}+ = {AC} = {A, C} So, No C. K.
  - ✓ {A, D}+ = {A, D} = {A, D, B} (Using D → B)

  = {A, D, B, C} (Using AB → CD) So, {A, D} is a C. K
  - ✓ {B, C}+ = {B, C}= {B, C, A} (Using C → A)

  = {A, D, B, C} (Using AB → CD) So, {B, C} is a C. K

Example 3: Finding Candidate Key

- Let R = (A, B, C, D) FD: {$AB \rightarrow CD$, $C \rightarrow A$, $D \rightarrow B$}
  - ✓ Continued..
  - ✓ {B, D}+ = {B, D} = {B, D} (Using $D \rightarrow B$) So, No C. K.
  - ✓ {C, D}+ = {C, D} = {C, D, A} (Using $C \rightarrow A$)

    = {A, D, B, C} (Using $D \rightarrow B$) So, {C, D} is a C. K

AC and BD are not candidate key. Now, again we combine them. But we consider only those which is not a super set of present candidate key.

In this case all are super set. So, {A, B}, {A, D}, {B, C}, {C, D} are candidate keys.

Total number of candidate keys: 4.

Example 3: Finding Super Key

- Let R = (A, B, C, D)   FD: {AB → CD, C → A, D → B}
  - ✓ Combining any attributes with candidate key becomes super key.
  - ✓ {A, B}, {A, D}, {B, C}, {C, D} are candidate keys.
  - ✓ So, Super keys are: {A, B}, {A, D}, {B, C}, {C, D} , {A, B, C}, {A, B, D}, {A, C, D}, {B, C, D}, {A, B, C, D}

Total number of super keys: 9.

Exercise: Finding Candidate Key and Super Key

- $R = (A, B, C, D, E)$ and FD: $\{AB \rightarrow CD, D \rightarrow A, BC \rightarrow DE\}$. Determine the total number of candidate keys and super keys.
- $R = (W, X, Y, Z)$ and FD: $\{Z \rightarrow W, Y \rightarrow XZ, XW \rightarrow Y\}$. Determine the total number of candidate keys and super keys.
- $R = (A, B, C, D, E, F)$ and FD: $\{AB \rightarrow C, DC \rightarrow AE, E \rightarrow F\}$. Determine the total number of candidate keys and super keys.
- $R = (A, B, C, D, E)$ and FD: $\{A \rightarrow BC, CD \rightarrow E, B \rightarrow D, E \rightarrow A\}$. Determine the total number of candidate keys and super keys.