for eg ① find the loan_no, branch_name & amount for loans of over Rs 1000

$\{ \langle l, b, a \rangle \mid \langle l, b, a \rangle \in loan \wedge a > 1200 \}$

② find all loan_no. for loans with an amount greater than 1200

$\{ \langle l \rangle \mid \exists\, b, a\, (\langle l, b, a \rangle \in loan \wedge a > 1200) \}$

⇒ Closure of a set F of functional dependencies.

↳ Closure of a set F of functional dependencies is the set of all functional dependencies logically implied by F.

We denote the closure of F by F^+

It can be inferred using Armstrong's Inference Rules

1) Reflexive rule :- if $\alpha \supseteq \beta$, then $\alpha \rightarrow \beta$
2) Augmentation rule :- If $\alpha \rightarrow \beta$, then $\alpha\gamma \rightarrow \beta\gamma$
3) transitive rule :- if $\alpha \rightarrow \beta$, $\beta \rightarrow \gamma$, then $\alpha \rightarrow \gamma$

The ~~correct~~ above three rules are Armstrong's Axioms

↳ Some additional rules, derivable from Armstrong Axioms :-

4) Union rule :- if $\alpha \rightarrow \beta$ and $\alpha \rightarrow \gamma$, then $\alpha \rightarrow \beta\gamma$
5) Decomposition rule :- if $\alpha \rightarrow \beta\gamma$, then $\alpha \rightarrow \beta$ and $\alpha \rightarrow \gamma$
6) Pseudotransitivity rule :- if $\alpha \rightarrow \beta$ and $\gamma\beta \rightarrow \delta$, then $\alpha\gamma \rightarrow \delta$

→ Candidate Keys

↳ Determining candidate keys from set of functional dependencies

↳ Rules to determine Candidate key :-

1) If an attribute is in none of the FDs, then it is in every candidate key

2) If an attribute occurs on the right side of an FD, but never occurs on the left hand side, then it is never in a candidate key.

3) If an attribute occurs on the left-hand side of an FD, but never occurs on the right hand side, then it is in every candidate key

4) If an attribute occurs both on the right hand side of an FD and left hand side of an FD, then one can not say anything about the attribute

↳ To find candidate key, identify which attribute are in each of the cases above. The ones in the first and third cases must be in every key. Call this set of attributes the Core. Compute the closure of Core. If closure of core determines all the attributes of relation, it is the candidate key. If closure of core don't determine all attributes, then some will be missing.

☆ One more rule to find exterior is

$\{R\} - \{(1)+(2)+(3)\}$

remaining attributes of R excluding (1), (2) & (3) rule attributes

Now attributes found in the (4) part are known as Exterior (attributes occurs both on the right side of an FD ~~[illegible]~~ and on the left hand side of FD). To get a candidate key one must add one or more exterior attributes to the core. Accordingly, add them to the core, first one at time, then two at a time, and so on, until every key has been found.

Eg $R = \{A, B, C, D\}$, $F = \{C \to A, B \to C\}$

Ans 1) D 2) A

3) B 4) C

Core $= \{BD\}$

$\{Core\}^+ \to \{BD\}^+ = \{B, C, A\}$

$\{D\}^+ = \{D\}$

because closure has all the attributes of relation, ∴ BD is the candidate key

Eg $R = \{A, B, C, D\}$; $F = \{C \to D, C \to A, B \to C\}$

Ans 1) None 2) DA

3) B 4) C

Core $= \{B\}$

$Core^+ = \{B\}^+ = \{C, A, D, B\}$

because closure has all the attributes of relation, ∴ B is the candidate key

Eg $R = \{A, B, C, D\}$, $F = \{ABC \to D,\ D \to A\}$

Ans 1) None 2) None

3) BC 4) ~~None~~ AD

Core = BC

$Core^{+} = \{B\}^{+} = \{B\}$

$\{C\}^{+} = \{C\}$

∵ BC do not determine all attributes of relation ∴ BC is not a candidate key

Exterior is AD

Adding one attribute at at time

$\{BCA\}^{+} = \{A, B, C, D\}$

$\{BCD\} = \{B C, D, A\}$

∴ ABC and BCD are candidate keys

$R = \{A, B, C, D\}$

Eg $F = \{AB \to C,\ AB \to D,\ C \to A,\ D \to B)$

Ans 1) None 2) None

3) None 4) ABCD

Core = Empty

Exterior = ABCD

finding closure of one attribute of Exterior at a time

$\{A\}^{+} = Empty$, $\{B\}^{+} = Empty$, $\{C\}^{+} = \{C, A\}$, $D^{+} = \{D, B\}$

Again no finding of all attributes of Relation

Now, taking two attributes at a time

Pair of attributes	Closure
AB	ABCD
AC	AC
AD	ABCD
BC	ABCD
BD	BD
CD	ABCD

So, AB, AD, BC and CD are candidate keys

⇒ Canonical Cover :-

A canonical cover F_c for F (a set of functional dependencies on a relational Schema) is a set of dependencies such that F logically implies all dependencies in FC, and F_c logically implies all dependencies in F

The set F_c has 2 important properties:-

1) No functional dependency in f_c contains an extraneous attribute

2) Each left side of a functional dependency in f_c is unique. That is, there are no two dependencies $a \to b$ and $c \to d$ in F_c such that $a = c$.

✡ Canonical cover F_c contains minimal set of dependencies equivalent to F, having no redundant dependencies or redundant part of dependencies.

✡ Algo on next page

Algo for computing Canonical cover

1) $F_c = F$

2) Repeat

i) Use the union rule to replace any dependency in F_c of the form $a \to b$ and $a \to d$ with $a \to bd$.

ii) Find a functional dependency in F_c with an extraneous attribute and delete it from F_c. i.e if ~~[illegible]~~ $\alpha \to B$ is functional dependency and there is extraneous attribute in α or in B, delete

3) Until F_c does not change

Eg $F = \{A \to B, \; B \to C, \; AB \to C\}$

~~F = {A→B, B→C, A→C, [illegible], AB→[illegible]}~~

F_C = from union of $A \to B$ and $AB \to C$
$A \to C$

remaing is $B \to C$

$\therefore F_C = \{A \to C, \; B \to C\}$

$\boxed{\text{Imp}}$

✡ For B+ tree & B tree index - look at the slides print out of Index Chapter
↳ few info. added in the slides manually