then

$i := i + 1;$

$R_i :=$ any candidate key for R;

8.29 /* Optionally, remove redundant relations */ of functional dependencies on the relation

repeat

if any schema R_j is contained in another schema R_k

then

/* Delete R_j */

$R_j := R_i;$

$i := i - 1;$

until no more R_js can be deleted

return $(R_1, R_2, \ldots, R_i)$

Figure 8.12 Dependency-preserving, lossless decomposition into 3NF.

$$A \rightarrow BCD$$
$$BC \rightarrow DE$$
$$B \rightarrow D$$
$$D \rightarrow A$$

d. Give a 3NF decomposition of r based on the canonical cover.

We've calculated canonical cover:

① $A \rightarrow BC$ ② $B \rightarrow DE$ ③ $D \rightarrow A$

Therefore, using

r1 (A, B, C). r2 (B. D, E). r3 (D. A)

Since F is not dependent on any attribute, and none of the attribute could determine F:

r4 (A. F) or r4 (D. F).

2、Consider a relation schema R(A,B,C,D,E) and its functional dependencies, F={ A→C, C→A, B→AC, D→AC }, complete the following questions:

a) Compute $(AD)^+$

b) Compute the candidate keys for R.

c) Compute the canonical cover *Fc*.

d) Is R in 3NF? If it is, justify your answer. If not, produce a decomposition of R into 3NF.(Ref. Figure 8.12)

let F_c be a canonical cover for F;

$i := 0;$

for each functional dependency $\alpha \rightarrow \beta$ in F_c

$i := i + 1;$

$R_i := \alpha\,\beta;$

if none of the schemas $R_j, j = 1, 2, \ldots, i$ contains a candidate key for R

then

$i := i + 1;$

$R_i :=$ any candidate key for R;

/* Optionally, remove redundant relations */

repeat

if any schema R_j is contained in another schema R_k

then

/* Delete R_j */

$R_j := R_i;$

$i := i - 1;$

until no more R_js can be deleted

return $(R_1, R_2, \ldots, R_i)$

Figure 8.12 Dependency-preserving, lossless decomposition into 3NF.

a) $(AD)^+ = \{AD\}$

① $A \rightarrow C$ $(AD)^+ = \{ADC\}$. ② $C \rightarrow A$ $(AD)^+ = \{ADC\}$. ④ $D \rightarrow AC$ $(AD)^+ = \{ADC\}$.

Therefore: $(AD)^+ = \{ADC\}$

b) $B \to AC$
$D \to AC$
$BD \to BD$
$\Rightarrow BD \to ABCD$, candidate key is BDE

E is dependent from ABCD.
B and D couldn't be concluded from each other.
Since A and C couldn't conclude B or D
Therefore, candidate key of R is BDE

c) Since $A \to C$ and $C \to A$
The F could be simplify as $A \to C$. $C \to A$. $B \to A$. $D \to A$
$F_C = \{A \to C. C \to A. B \to A. D \to A\}$

d) It's not 3NF, since $A \to C$ and $C \to A$

$r_1(B.D.A)$. $r_2(A, C)$

3. Suppose that we have a schema ***R(A, B, C, D, E)***. You are given the following dependencies:

$$A \to B$$
$$BC \to E$$
$$ED \to A$$

a) List all candidate keys for R.
b) Is *R* in 3NF? If it is, justify your answer. If not, produce a decomposition of *R* into 3NF. (Ref. Figure 8.12)

a) ① $A \to B$
$A^+ = (AB)$. A couldn't conclude all the attributes.

② $BC \to E$
$(AC)^+ = (ABCE)$ D couldn't be derived.

③ $ED \to A$
$(ACD)^+ = (ABCDE)$ ACD is one of the candidate..
$(ED)^+ = (ABED)$ ED couldn't derive C.

Therefore: candidate keys are: ACD. CDE

b) Since: ACD → BCD

BCD → ED

ED → A

Therefore, R is not 3NF.

Decomposition r_1 (A.B.C.E).

r_2 (E.D.A)

4. Which normal form do the schemas you obtained in Chapter 7 Exercise belong to? Is it possible to transform them into 3NF? And how?

2NF

It's possible to transform.

Here are the steps:

① Identify Functional Dependencies.

② Check Transitive Dependencies

③ Remove Transitive Dependencies

④ Update Relationship til there exists no transitive Dependencies