Database Management System (DBMS)

Lecture-34

Dharmendra Kumar

November 8, 2020

Boyce-codd normal form (BCNF)

A relation schema R is in BCNF with respect to a set F of functional dependencies if, for all functional dependencies in F^+ of the form $\alpha \rightarrow \beta$, where $\alpha \subseteq R$ and $\beta \subseteq R$, at least one of the following holds:

- $\alpha \rightarrow \beta$ is a trivial functional dependency.
- α is a super key for R.

Example: The relation schema Student_Performance (name, courseNo, rollNo, grade) has the following FDs:

name,courseNo $\rightarrow$ grade

rollNo,courseNo $\rightarrow$ grade

name $\rightarrow$ rollNo

rollNo $\rightarrow$ name

Is this relation in BCNF?

Solution:

Consider FD, name,courseNo $\rightarrow$ grade. Clearly, $\{$name, courseNo$\}$ is a super key, therefore this satisfy 2nd criteria of BCNF.

Consider FD, rollNo,courseNo $\rightarrow$ grade. Clearly, $\{$rollNo, courseNo$\}$ is a super key, therefore this satisfy 2nd criteria of BCNF.

Consider FD, name $\rightarrow$ rollNo. Clearly, this functional dependency not satisfy any condition of BCNF. Therefore, this is not in BCNF.

Example: Consider the following relation schemas and their respective functional dependencies:

Customer = (customer-name, customer-street, customer-city)

customer-name → customer-street, customer-city

Branch = (branch-name, assets, branch-city)

branch-name → assets, branch-city

Loan = (branch-name, customer-name, loan-number, amount)

loan-number → amount, branch-name

Clearly, Customer and Branch schema are in BCNF, because left side of functional dependency **customer-name → customer-street, customer-city** and **branch-name → assets, branch-city** is super key.

But Loan schema is not in BCNF, because left side of functional dependency **loan-number → amount, branch-name** is not super key and this functional dependency is also not trivial.

Decomposition into 2NF

Example:

Let R = { A, B, C, D} and F = { AB $\rightarrow$ C, B $\rightarrow$ D }.

Is this relation schema in 2NF? If not then decompose it into 2NF.

Solution:

Here, Primary key = {A,B}.

Clearly, this is not in 2NF because partial dependency holds.

Now, we decompose R into R_1 and R_2 as the following:-

R_1 = (A, B, C), F_1 = {AB $\rightarrow$ C}

R_2 = (B, D), F_2 = {B $\rightarrow$ D}

Now, R_1 and R_2 are in 2NF.

Decomposition into Normal form

Example:

Student-course-info(Name, Course, Grade, Phone-no, Major, Course-dept)

F = { Name $\rightarrow$ Phone-no Major, Course $\rightarrow$ Course-dept, Name Course $\rightarrow$ Grade }

Is this relation schema in 2NF? If not then decompose it into 2NF.

Solution:

Here, Primary key = {Name,Course}.

Clearly, this is not in 2NF because partial dependency holds.

Now, we decompose R into R_1, R_2 and R_3 as the following:-

R_1 = (Name, Phone-no, Major), F_1 = {Name $\rightarrow$ Phone-no Major}

R_2 = (Course, Course-dept), F_2 = {Course $\rightarrow$ Course-dept}

R_3 = (Name, Course, Grade), F_3 = {Name Course $\rightarrow$ Grade}

Now, R_1, R_2 and R_3 are in 3NF.

Decomposition into 3NF

Example:

Let R = { A, B, C, D} and F = { A $\rightarrow$ B, A $\rightarrow$ C, B $\rightarrow$ D }.

Is this relation schema in 3NF? If not then decompose it into 3NF.

Solution:

Here, Primary key = {A}.

Clearly, this is not in 3NF because transitivity dependency holds.

Now, we decompose R into R_1 and R_2 as the following:-

R_1 = (A, B, C), F_1 = {A $\rightarrow$ B, A $\rightarrow$ C}

R_2 = (B, D), F_2 = {B $\rightarrow$ D}

Now, R_1 and R_2 are in 3NF.

Decomposition into Normal form

Example:

Let Banker-info = { branch-name, customer-name, banker-name, office-number} and F = { banker-name $\rightarrow$ branch-name office-number, customer-name branch-name $\rightarrow$ banker-name}.

Is this relation schema in 3NF? If not then decompose it into 3NF.

Solution:

Here, Primary key = {customer-name, branch-name}.

Clearly, this is not in 3NF because transitivity dependency holds.

Now, we decompose relation Banker-info into R_1 and R_2 as the following:-

R_1 = (banker-name, branch-name, office-number), F_1 = {banker-name $\rightarrow$ branch-name office-number}

R_2 = (customer-name, branch-name, banker-name), F_2 = {customer-name branch-name $\rightarrow$ banker-name}

Now, R_1 and R_2 are in 3NF.

Decomposition into Normal form

Decomposition into BCNF

If R is not in BCNF, then we can decompose R into a collection of BCNF schemas $R_1, R_2, ..., R_n$ by the algorithm. The decomposition that the algorithm generates is not only in BCNF, but is also a lossless-join decomposition.

Input: R and F

Output: result

result $\leftarrow R$

done $\leftarrow$ *false*

compute F^+

while *(done = false)* **do**

  if *(there is a schema R_i in result that is not in BCNF)* **then**

    let $\alpha \rightarrow \beta$ be a nontrivial functional dependency that holds on R_i such that $\alpha \rightarrow R_i$ is not in F^+, and $\alpha \cap \beta = \phi$

    $result \leftarrow (result - R_i) \cup (R_i - \beta) \cup (\alpha, \beta)$

  end

  else

    done = true

  end

end

Decomposition into Normal form

Example:

Consider the following schema:-

Lending-schema = (branch-name, branch-city, assets, customer-name, loan-number, amount)

F = { branch-name $\rightarrow$ assets branch-city, loan-number $\rightarrow$ amount branch-name}

Is this relation schema in BCNF? If not then decompose it into BCNF.

Decomposition into Normal form

Solution:

Here, Primary key = {customer-name, loan-number}.

Clearly, this is not in BCNF.

Now, we decompose relation Lending-schema into R_1 and R_2 as the following:-

$R_1 =$ (customer-name, loan-number, amount, branch-name)

$F_1 = \{$ loan-number $\rightarrow$ amount branch-name$\}$

$R_2 =$ (branch-name, assets, branch-city)

$F_2 = \{$ branch-name $\rightarrow$ assets branch-city$\}$

Clearly, R_1 is not in BCNF. Therefore, we again decompose R_1.

Now, we decompose relation R_1 into R_3 and R_4 as the following:-

$R_3 =$ (customer-name, loan-number)

$F_3 = \phi$

$R_4 =$ (loan-number, amount, branch-name)

$F_4 = \{$ loan-number $\rightarrow$ amount branch-name$\}$

Now, the final relation schema are R_2, R_3 and R_4. All these are in BCNF.