

## ● Normalization proofs of Relations:

### (1) **LoggedUser:**

⇒ Attributes:

LoggedUser{UID, UName, Email, PhoneNo, Password,  
Gender, DOB, UBlockStatus, DID}

⇒ Functional dependencies:

UID -> UName  
UID -> Email  
UID -> PhoneNo  
UID -> Password  
UID -> Gender  
UID -> DOB  
UID -> UBlockStatus  
UID -> DID

Here Closure,

UID+ = {UID, UName, Email, PhoneNo, Password, Gender, DOB,  
UBlockStatus, DID}

So, **Primary key = UID**

The left side of all FDs in minimal set of FDs for the relation "LoggedUser" is UID, which is primary key of the relation "LoggedUser". There is no other key which can determine the prime attribute (UID) so we can say the relation is in BCNF.

## (2) UnloggedUser:

⇒ Attributes:

UnloggedUser {UnID}

**Primary key = UnID**

All attributes can be determined by the primary key UnID so the relation “UnloggedUser” is in BCNF.

## (3) Advertisement:

⇒ Attributes:

Advertisement {AdID, AdvertisementCost, PaymentPerView, DID}

⇒ Functional dependencies:

AdID → AdvertisementCost

AdID → PaymentPerView

AdID → DID

Here Closure,

AdID<sup>+</sup> = {AdID, AdvertisementCost, PaymentPerView, DID}

So, **Primary key = AdID**

The left side of all FDs in minimal set of FDs for the relation “Advertisement” is AdID, which is primary key of the relation

“Advertisement”. There is no other key which can determine the prime attribute (AdID) so we can say the relation is in BCNF.

#### **(4) Likes\_1:**

⇒ Attributes:

Likes\_1{LikesID, PostID, UID}

⇒ Functional dependencies:

LikesID → PostID

LikesID → UID

Here Closure,

LikesID<sup>+</sup> = {LikesID, PostID, UID}

So, **Primary key = LikesID**

The left side of all FDs in minimal set of FDs for the relation “Likes\_1” is LikesID, which is primary key of the relation “Likes\_1”. There is no other key which can determine the prime attribute (LikesID) so we can say the relation is in BCNF.

#### **(5) Likes\_2:**

⇒ Attributes:

Likes\_2{LikesID, PPostID, UID}

⇒ Functional dependencies:

LikesID  $\rightarrow$  PPostID

LikesID  $\rightarrow$  UID

Here Closure,

LikesID  $\rightarrow$  {LikesID, PPostID, UID}

So, **Primary key = LikesID**

The left side of all FDs in minimal set of FDs for the relation "Likes\_2" is LikesID, which is primary key of the relation "Likes\_2". There is no other key which can determine the prime attribute (LikesID) so we can say the relation is in BCNF.

**Note:** At start we had only one relation Likes for. But it had PostID and PPostID both as their attributes but one of them had to be NULL all the time. So, to remove that redundancy we created 2 different relations for Likes.

## **(6) Report\_1:**

$\Rightarrow$  Attributes:

Report\_1{ReportID, Rtype, Rtime, DID, PostID, UID}

$\Rightarrow$  Functional dependencies:

ReportID  $\rightarrow$  Rtype

ReportID  $\rightarrow$  Rtime

ReportID  $\rightarrow$  DID

ReportID  $\rightarrow$  PostID

ReportID  $\rightarrow$  UID

Here Closure,

$\text{ReportID}^+ = \{\text{ReportID}, \text{Rtype}, \text{Rtime}, \text{DID}, \text{PostID}, \text{UID}\}$

So, **Primary key = ReportID**

The left side of all FDs in minimal set of FDs for the relation "Report\_1" is ReportID, which is primary key of the relation "Report\_1". There is no other key which can determine the prime attribute (ReportID) so we can say the relation is in BCNF.

## (7) Report\_2:

⇒ Attributes:

$\text{Report}_2\{\text{ReportID}, \text{Rtype}, \text{Rtime}, \text{DID}, \text{PPostID}, \text{UID}\}$

⇒ Functional dependencies:

$\text{ReportID} \rightarrow \text{Rtype}$

$\text{ReportID} \rightarrow \text{Rtime}$

$\text{ReportID} \rightarrow \text{DID}$

$\text{ReportID} \rightarrow \text{PPostID}$

$\text{ReportID} \rightarrow \text{UID}$

Here Closure,

$\text{ReportID}^+ = \{\text{ReportID}, \text{Rtype}, \text{Rtime}, \text{DID}, \text{PPostID}, \text{UID}\}$

So, **Primary key = ReportID**

The left side of all FDs in minimal set of FDs for the relation "Report\_2" is ReportID, which is primary key of the relation "Report\_2". There is no other key which can determine the prime attribute (ReportID) so we can say the relation is in BCNF.

**Note:** At start we had only one relation Report for. But it had PostID and PPostID both as their attributes but one of them had to be NULL all the time. So, to remove that redundancy we created 2 different relations for Report.

## **(8) AdministrationDepartment:**

⇒ Attributes:

AdministrationDepartment {DID, Dname}

⇒ Functional dependencies:

DID → Dname

Here Closure,

$DID^+ = \{DID, Dname\}$

So, **Primary key = DID**

The left side of all FDs in minimal set of FDs for the relation "AdministrationDepartment" is DID, which is primary key of the relation "AdministrationDepartment". There is no other key which can determine the prime attribute (DID) so we can say the relation is in BCNF.

## (9) Publisher:

⇒ Attributes:

Publisher{PublisherID, PublisherName, Email, PhoneNo, Address}

⇒ Functional dependencies:

PublisherID → PublisherName

PublisherID → Email

PublisherID → PhoneNo

PublisherID → Address

Here Closure,

$\text{PublisherID}^+ = \{\text{PublisherID}, \text{PublisherName}, \text{Email}, \text{PhoneNo}, \text{Address}\}$

So, **Primary key = PublisherID**

The left side of all FDs in minimal set of FDs for the relation "Publisher" is PublisherID, which is primary key of the relation "Publisher". There is no other key which can determine the prime attribute (PublisherID) so we can say the relation is in BCNF.

## (10) PublisherPost:

⇒ Attributes:

PublisherPost{PPostID, Title, PostTime, PBlocktatus, UpdateTime, Genre, Category, DID, PublisherID}

⇒ Functional dependencies:

PPostID → Title  
PPostID → PostTime  
PPostID → PBlockstatus  
PPostID → UpdateTime  
PPostID → Genre  
PPostID → Category  
PPostID → DID  
PPostID → PublisherID

Here Closure,

PPostID + = {PPostID, Title, PostTime, PBlocktatus, UpdateTime, Genre, Category, DID, PublisherID}

So, **Primary key = PPostID**

The left side of all FDs in minimal set of FDs for the relation “PublisherPost” is PPostID, which is primary key of the relation “PublisherPost”. There is no other key which can determine the prime attribute (PPostID) so we can say the relation is in BCNF.

## **(11) Share\_1:**

⇒ Attributes:

Share\_1{ShareID, PostID, UID}

⇒ Functional dependencies:

ShareID → PostID



ShareID  $\rightarrow$  UID

Here Closure,

ShareID<sup>+</sup> = {ShareID, PostID, UID}

So, **Primary key = ShareID**

The left side of all FDs in minimal set of FDs for the relation "Share\_1" is ShareID, which is primary key of the relation "Share\_1". There is no other key which can determine the prime attribute (ShareID) so we can say the relation is in BCNF.

## **(12) Share\_2:**

$\Rightarrow$  Attributes:

Share\_2{ShareID, PPostID, UID}

$\Rightarrow$  Functional dependencies:

ShareID  $\rightarrow$  PPostID

ShareID  $\rightarrow$  UID

Here Closure,

ShareID<sup>+</sup> = {ShareID, PostID, UID}

So, **Primary key = ShareID**

The left side of all FDs in minimal set of FDs for the relation "Share\_2" is ShareID, which is primary key of the relation "Share\_2". There is no other key which can determine the prime attribute (ShareID) so we can say the relation is in BCNF.

**Note:** At start we had only one relation for Share. But it had PostID and PPostID both as their attributes but one of them had to be NULL all the time. So, to remove that redundancy we created 2 different relations for Share.

### **(13) Comment\_1:**

⇒ Attributes:

Comment\_1 {CommentID, PostID, UID, Description}

⇒ Functional dependencies:

CommentID → PostID

CommentID → UID

CommentID → Description

Here Closure,

CommentID<sup>+</sup> = {CommentID, PostID, UID, Description}

So, **Primary key = CommentID**

The left side of all FDs in minimal set of FDs for the relation "Comment\_1" is CommentID, which is primary key of the relation "Comment\_1". There is no other key which can determine the prime attribute (CommentID) so we can say the relation is in BCNF.

### **(14) Comment\_2:**

⇒ Attributes:

Comment\_2 {CommentID, PPostID, UID, Description}

⇒ Functional dependencies:

CommentID → PPostID

CommentID → UID

CommentID → Description

Here Closure,

CommentID<sup>+</sup> = {CommentID, PPostID, UID, Description}

So, **Primary key = CommentID**

The left side of all FDs in minimal set of FDs for the relation

“Comment\_2” is CommentID, which is primary key of the relation

“Comment\_2”. There is no other key which can determine the prime attribute (CommentID) so we can say the relation is in BCNF.

**Note:** At start we had only one relation Comment for. But it had PostID and PPostID both as their attributes but one of them had to be NULL all the time. So, to remove that redundancy we created 2 different relations for Comment.

## **(15) UserPost:**

⇒ Attributes:

UserPost {PostID, Title, PostTime, PBlockstatus, Genre, Category, DID, UID}

⇒ Functional dependencies:

PostID → Title

PostID → Post time

PostID → PBlock\_status

PostID → Genre

PostID → Category

PostID → DID

PostID → UID

Here Closure,

PostID<sup>+</sup> = {PostID, Title, Post time, PBlock\_status, Genre, Category, DID, UID}

So, **Primary key = PostID**

The left side of all FDs in minimal set of FDs for the relation "UserPost" is PostID, which is primary key of the relation "UserPost". There is no other key which can determine the prime attribute (PostID) so we can say the relation is in BCNF.

## **(16) Staff:**

⇒ Attributes:

Staff {StaffID, Staffname, Address, DOB, DID, MgrID}

⇒ Functional dependencies:

StaffID  $\rightarrow$  Staffname

StaffID  $\rightarrow$  Address

StaffID  $\rightarrow$  DOB

StaffID  $\rightarrow$  DID

StaffID  $\rightarrow$  MgrID

Here Closure,

StaffID<sup>+</sup> = {StaffID, Staffname, Address, DOB, DID, MgrID}

So, **Primary key = StaffID**

The left side of all FDs in minimal set of FDs for the relation "Staff" is StaffID, which is primary key of the relation "Staff". There is no other key which can determine the prime attribute (StaffID) so we can say the relation is in BCNF.

### **(17) View\_Un\_1:**

$\Rightarrow$  Attributes:

View\_Un\_1{ViewTime, PostID, UnID}

**Primary key = {ViewTime,PostID,UnID}**

All attributes can be determined by the primary key {ViewTime,PostID,UnID} so the relation "View\_Un\_1" is in BCNF.

### **(18) View\_Un\_2:**

$\Rightarrow$  Attributes:

View\_Un\_2{ViewTime, PPostID, UnID}

**Primary key = {ViewTime,PPostID,UnID}**

All attributes can be determined by the primary key {ViewTime,PPostID,UnID} so the relation “View\_Un\_2” is in BCNF.

### **(19) View\_L\_1:**

⇒ Attributes:

View\_L\_1{ViewTime, PostID, UID}

**Primary key = {ViewTime,PostID,UID}**

All attributes can be determined by the primary key {ViewTime, PostID, UID} so the relation “View\_L\_1” is in BCNF.

### **(20) View\_L\_2:**

⇒ Attributes:

View\_L\_1{ViewTime, PPostID, UID}

**Primary key = {ViewTime,PPostID,UID}**

All attributes can be determined by the primary key {ViewTime, PPostID, UID} so the relation “View\_L\_2” is in BCNF.

**Note:** At start we had two different relations view1 and view2 for logged and unlogged user respectively. But it had PostID and PPostID both as their attributes but one of them had to be NULL all the time. So, to remove that redundancy we created 4 different relations for view.

### **(21) CanReport\_1:**

⇒ Attributes:

CanReport\_1{PostID, UID}

**Primary key = {PostID, UID}**

All attributes can be determined by the primary key {PostID, UID} so the relation “CanReport\_1” is in BCNF.

### **(22) CanReport\_2:**

⇒ Attributes:

CanReport\_2{PPostID, UID}

**Primary key = {PPostID, UID}**

All attributes can be determined by the primary key {PPostID, UID} so the relation “CanReport\_2” is in BCNF.

**Note:** At start we had only one relation for CanReport. But it had PostID and PPostID both as their attributes but one of them had to be NULL all the time. So, to remove that redundancy we created 2 different relations for CanReport.

### **(23) Follow:**

⇒ Attributes:

Follow {UID1, UID2}

**Primary key = {UID1, UID2}**

All attributes can be determined by the primary key {UID1, UID2} so the relation “Follow” is in BCNF.

### **(24) Follow\_2:**

⇒ Attributes:

Follow\_2 {UID, PublisherID}

**Primary key = {UID, PublisherID}**

All attributes can be determined by the primary key {UID, PublisherID} so the relation “Follow\_2” is in BCNF.



**Note:** At start we had only one relation for follow. But it had PostID and PPostID both as their attributes but one of them had to be NULL all the time. So, to remove that redundancy we created 2 different relations for follow.

### **(25) DealWith:**

⇒ Attributes:

DealWith {DID, PublisherID}

**Primary key = {DID, PublisherID}**

All attributes can be determined by the primary key {DID, PublisherID} so the relation “DealWith” is in BCNF.

### **(26) Contains:**

⇒ Attributes:

Contains {PPostID, AdID}

**Primary key = {PPostID, AdID}**

All attributes can be determined by the primary key {PPostID, AdID} so the relation “Contains” is in BCNF.

### **(27) Author:**

⇒ Attributes:

Author {AuthorName, PPostID}

**Primary key = {AuthorName, PPostID}**

All attributes can be determined by the primary key {AuthorName, PPostID} so the relation "Author" is in BCNF.