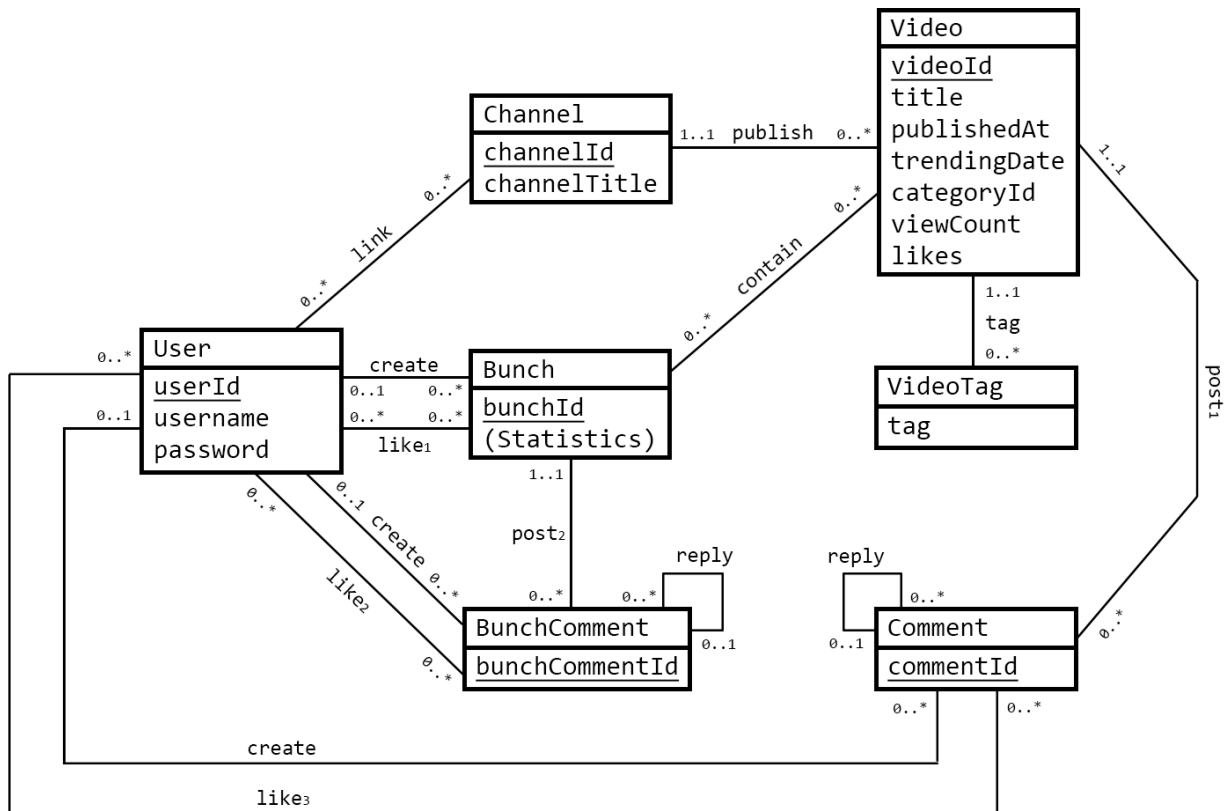


CS 411 PT1 Stage 2

Oct. 2, 2023

Team #43 InfelPhira

① UML diagram



Channel: store the YouTube channel information.

Video: store the YouTube video data (tag exclusive).

VideoTag: store the YouTube video tags.

User: store the user account information.

Bunch: store the user-created video bunch statistics.

Comment: store the user comments on the videos.

BunchComment: store the user comments on the bunches.

② BCNF

The **bold font** is the titles of tables, and below are the attributes.

The underlined attributes are the primary keys.

The *italic attributes* are the foreign keys.

Channel	Video	Bunch	User	Comment	BunchComment	BunchContain	ChannelLink	BunchLike	CommentLike	BunchCommentLike	VideoTag
<u>ChannelId</u>	<u>VideoId</u>	<u>BunchId</u>	<u>UserId</u>	<u>CommentId</u>	<u>BunchCommentId</u>	<u>VideoId</u>	<u>UserId</u>	<u>UserId</u>	<u>UserId</u>	<u>UserId</u>	<u>VideoId</u>
ChannelTitle	Title	<i>UserId</i>	Username	<i>UserId</i>	<i>UserId</i>	<i>BunchId</i>	<i>ChannelId</i>	<i>BunchId</i>	<i>CommentId</i>	<i>BunchCommentId</i>	Tag
	PublishedAt		Password	<i>VideoId</i>	<i>BunchId</i>						
	TrendingDate			<i>ParentCommentId</i>	<i>ParentBunchCommentId</i>						
	CategoryId										
	ViewCount										
	Likes										
	<i>ChannelId</i>										

All of the above tables meet the following criteria:

- (1) Each attribute in the table is not separable.
- (2) Non-primary attributes must depend on the entire primary key or candidate key.
- (3) Any non-primary attribute cannot depend on other non-primary attributes.
- (4) Unable to use BCNF Decomposition.

So, the tables above are in BCNF.

BCNF vs 3NF:

BCNF has less redundancy, but the number of tables might be greater than that in 3NF. In this case, the structure is both in BCNF and 3NF.

③ Relational schema

Channel (ChannelId: INT [PK], ChannelTitle: VARCHAR(255))

Video (VideoId: INT [PK], Title: VARCHAR(255), PublishedAt: DATETIME, TrendingDate: DATETIME, CategoryId: INT, ViewCount: INT, Likes: INT, ChannelId: INT [FK to Channel.ChannelId])

Bunch (BunchId: INT [PK], UserId: INT [FK to User.UserId])

User (UserId: INT [PK], Username: VARCHAR(255), Password: INT)

Comment (CommentId: INT [PK], UserId: INT [FK to User.UserId], VideoId: INT [FK to Video.VideoId], ParentCommentID: INT)

BunchComment (BunchCommentId: INT [PK], UserId: INT [FK to User.UserId], BunchId: INT [FK to Bunch.BunchId], ParentBunchCommentID: INT)

BunchContain (VideoId: INT [PK1, FK to Video.VideoId], BunchId: INT [PK2, FK to Bunch.BunchId])

ChannelLink (UserId: INT [PK1, FK to User.UserId], ChannelId: INT [PK2, FK to Channel.ChannelId])

BunchLike (UserId: INT [PK1, FK to User.UserId], BunchId: INT [PK2, FK to Bunch.BunchId])

CommentLike (UserId: INT [PK1, FK to User.UserId], CommentId: INT [PK2, FK to Comment.CommentId])

BunchComentLike (UserId: INT [PK1, FK to User.UserId], BunchCommentId: INT [PK2, FK to BunchComment.BunchCommentId])

VideoTag (VideoId: INT [FK to Video.VideoId], Tag: VARCHAR(255))