INFO90002-Assignment 1
“HealthTalk” DataBase

Runze Li	672450	runzel
Mingyu Gao	692634	mingyug
Liang Han	712397	lingh2

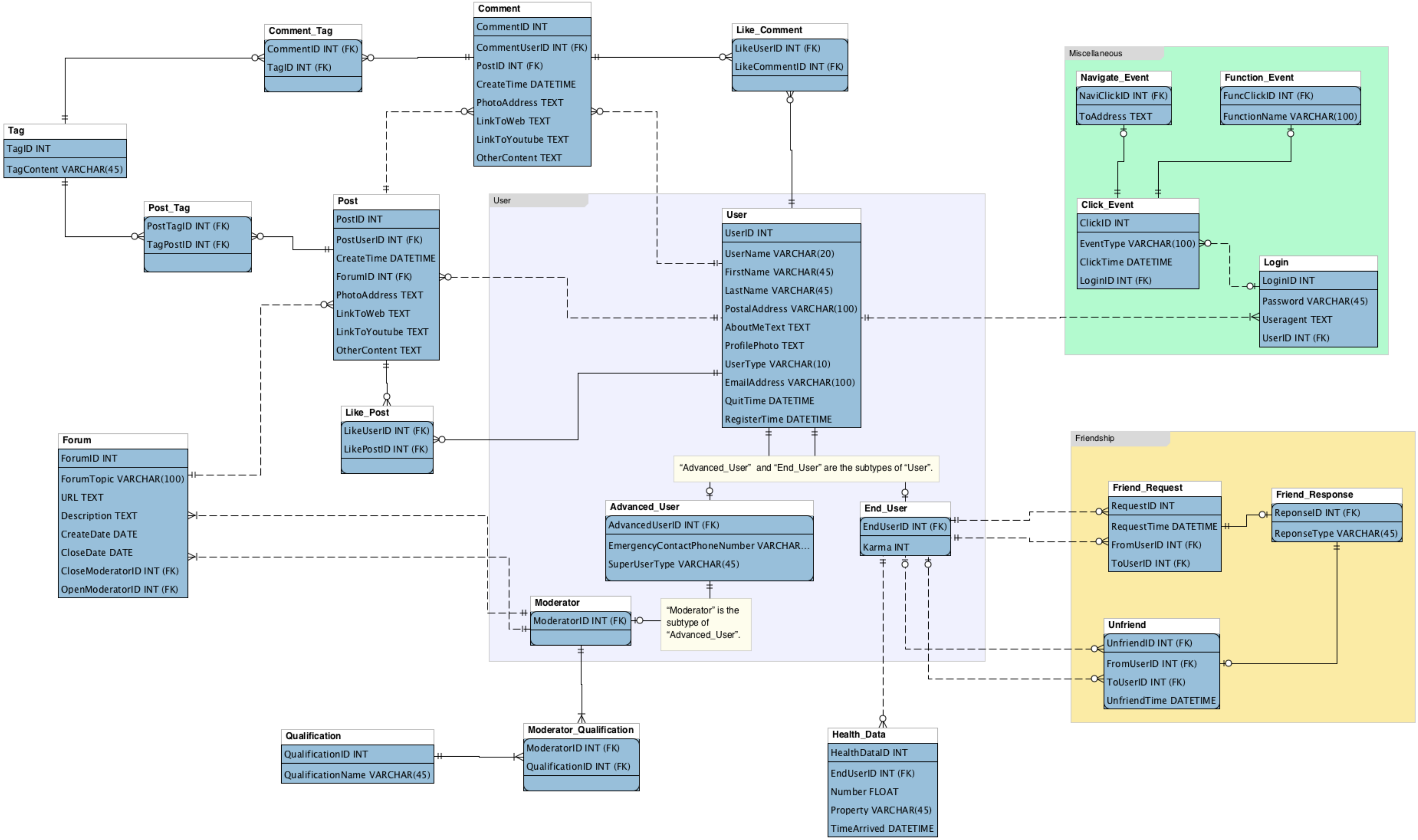
Content

INFO90002-Assignment 1 -“HealthTalk” DataBase
Team Members: **Runze Li** - 672450 – runzel , **Mingyu Gao** – 692634 - mingyug, **Liang Han** – 712397- lingh2

Part 1 Assumptions

1. **The "User" including “super_user” and “end_user” has “usertype” attribute**. It means the “usertype”(“Newbine”, “Regular” and “Leader”) existing in the “User” entity.
2. **The time one user account is created, it performs one 'login' action.** So the relationship between “User” and “Login” is 1 to “1 or many” (not “0 or many”).
3. **Only the “click” actions after login can be recorded in the database.** So the records in the “Click_Event” entity is designed to be corresponded to each login record in “Login” Entity.
4. **Each moderator has at least one qualification.** So the relationship between “Moderator” and “Moderator_Qualification” is 1 to “1 or many”.
5. **The possibility that each comment or the post has the same content of photo, link to web, YouTube or other content is so low that we can ignore.** So all these attributes are added in both of the “Comment” and “Post” respectively. There will be no difference between this design and creating isolated entities to store photo (address) or link based on this assumption.
6. **If two users terminate the friend relationship between them. They two still can send friend request to each other again.** This means that the number of records in the “Friend_Request” entity with the same “FromUser” and “ToUser” can be more than one. So there is isolated primary key “RequestID” in the “Friend_Request” instead of the composite primary key made up of the “FromUserID” and “ToUserID”.

Part 2 Physical Model



Part 3 Dictionary

List of entities

Entities	Entity Type	Primary Key	#Attributes
User	Independent	UserID	11
Advanced_User	Dependent	AdvancedUserID	3
End_User	Dependent	EndUserID	2
Moderator	Dependent	ModeratorID	1
Qualification	Independent	QualificationID	2
Moderator_Qualification	Dependent	ModeratorID,QualificationID	2
Friend_Request	Independent	RequestID	4
Friend_Response	Dependent	ResponseID	2
Unfriend	Dependent	UnfirendID	4
Forum	Independent	ForumID	8
Post	Independent	PostID	8
Comment	Independent	CommentID	8
Like_Comment	Dependent	LikeUserID,LikeCommentID	2
Like_Post	Dependent	LikeUserID,LikePostID	2
Tag	Independent	TagID	2
Post_Tag	Dependent	PostTagID,PostID	2
Comment_Tag	Dependent	CommentTagID,CommentID	2
HealthData	Independent	HealthDataID	5
Login	Independent	LoginID	4
Click_Event	Independent	ClickID	4
Navigate_Event	Dependent	NaviClickID	5
Function_Event	Dependent	FuncClickID	2

Entity 'User'

Description

This entity represents users who have registered to use this application and each user has an id.

Attributes

Key	Attribute	Data Type	Not Null	Unique	Description
PK	UserID	INT	YES	NO	This primary key is unique integer. It is generated when a new user is created. E.g. "1","2","3",etc.
	UserName	VARCHAR(20)	YES	YES	The name of the user and it is up to 20 characters. E.g. "happylee", "mooncake",etc.
	FirstName	VARCHAR(45)	YES	NO	The real first name of the user. It is up to 45 characters. E.g. "Jordan", "Howard",etc.
	LastName	VARCHAR(45)	YES	NO	The real last name of the user. E.g.

					"Andy", "Steve",etc.
	PostalAddress	VARCHAR(100)	YES	NO	The post address that the user provides and it is up to 100 characters. E.g. "110 Lygon Street, Carlton VIC 3053"
	AboutMeText	TEXT	YES	NO	The paragraph of "about me "text describes the information of user in short sentences.
	ProfilePhoto	TEXT	YES	NO	The profile photo is provided by a user when the user is registering to this application. The photo can be stored in the format of file in the DBMS.
	UserType	VARCHAR(10)	YES	NO	Each user has the attribute of type and it is up to10 characters. New users are created with type "newbie"; after a month of regular use they become "regular"; after a year, "leader".
	EmailAddress	VARCHAR(100)	YES	NO	This email address is provided by the user and stored in the DBMS up to 100 characters. E.g. abcdefg@gmail.com.
	QuitTime	DATETIME	NO	NO	The QuitTime is generated when a user quits the application, and the time will be recorded in this DBMS.
	RegisterTime	DATETIME	YES	NO	The RegisterTime is recorded when a user is registering to become the user of this application. It is stored in the DATETIME type in the DBMS.

Entity 'Advanced_User'

Description

This entity is subtype of the entity "User", it inherits the attributes of the User. And it is generated with specific ID and relevant attributes.

Attributes

Key	Attribute	Data Type	Not Null	Unique	Description
PFK	AdvancedUserID	INT	YES	NO	The UserID from Entity User makes up both primary key and foreign key in the entity Advanced_User as AdvancedUserID. It is unique. E.g. "1", "2", "3".
	EmergencyContactPhoneNumber	VARCHAR(20)	YES	NO	The advanced user has emergency contact phone number which should be stored in the DBMS up to 20 characters. E.g. "0418477983", "0439873678".
	SuperUserType	VARCHAR(45)	YES	NO	This describe the type of the advanced user, and it will be granted with type name. E.g. "Moderator", "Administrator".

Entity 'End_User'

Description

"End_User" is the subtype of the entity "User". This entity is set to store the normal users of this application.

Attributes

Key	Attribute	Data Type	Not Null	Unique	Description
PFK	EndUserID	INT	YES	YES	A foreign key from Entity "User" makes up the primary key of this Entity. It is unique.

	Kama	INT	YES	NO	This attribute builds in integer and it describes the content that the user contributes quality content to the system. E.g. "20", "30".

Entity 'Moderator'

Description

This "Moderator" Entity is the subtype of Entity "User". And the primary key of this entity is from the entity "Advanced_User". If the user is "moderator", corresponding data will be associated with this entity.

Attributes

Key	Attribute	Data Type	Not Null	Unique	Description
PFK	ModeratorID	INT	YES	YES	A foreign key from entity of "Advanced_User" makes up the primary key. It is generated when a "moderator" user is registered and it unique.

Entity 'Moderator_Qualification'

Description

This entity is the Associative Entity of "Moderator" and "Qualification". It holds the ModeratorID and QualificationID both as the primary key. This means that one moderator can have more than one qualifications and each qualifications is stored in the entity.

Attributes

Key	Attribute	Data Type	Not Null	Unique	Description
PFK	ModeratorID	INT	YES	NO	A foreign key from entity "Moderator" makes up the primary key.
PFK	QualificationID	INT	YES	NO	A foreign key from entity "Qualification" makes up the primary key.

Entity 'Qualification'

Description

This entity stores the information of different types of qualifications that "moderator" user owns.

Attributes

Key	Attribute	Data Type	Not Null	Unique	Description
PK	QualificationID	INT	YES	YES	This is the primary key of the entity. It is unique. It is generated when a new qualification type is recorded.
	QualificationName	VARCHAR(45)	YES	YES	This describes the qualification type which "moderator" user can hold. It is up to 45 characters and one "moderator" user can have one or more qualification types. E.g. "Ph.D", "M.D", etc.

Entity 'Health_Data'

Description

This entity has a record for all the health data of users, a user can store corresponding health information like how far the user has run, calories burned, time spent asleep, and blood pressure.

Attributes

Key	Attribute	Data Type	Not Null	Unique	Description
PK	HealthDataID	INT	YES	YES	This is primary key and it unique. It is generated when a new health data arrives.
FK	EndUserID	INT	NO	YES	A foreign key from entity "End_User". It records to whom the each record of health data belong.
	Number	FLOAT	YES	NO	This records the value of the health data type. It is a float number.
	Property	VARCHAR(45)	YES	NO	This describes the type of health information. It consists of the types like user's running distance, calories burned, time spent asleep, and blood pressure. It is unique.
	TimeAddrived	DATETIME	YES	NO	This describes when the health data arrives and stored in the DATETIME format in DBMS.

Entity 'Friend_Request'

Description

This entity records the request of becoming a friend between two different users. It has two foreign keys "FromUserID" and "ToUserID" referring to the "End_User" entity. The combination of the "FromUserID" and "ToUserID" should have been regarded as a composite primary key. But there is a scenario where two users "unfriend" and then "befriend" again. What's more, "Friend_Response" is needed to record the response of each sending request, so it needs another attribute "RequestID" which is used as a primary key.

Attributes

Key	Attribute	Data Type	Not Null	Unique	Description
PK	RequestID	INT	YES	YES	This is the unique attribute which records different requests from users. It is used as a reference of the Response entity.
FK	FromUserID	INT	YES	NO	A foreign-key, coming from the entity End_User. It records who arise the request to become a friend.
FK	ToUserID	INT	YES	NO	A foreign-key, coming from the entity End_User. It records who receives the request of being a friend.
	RequestTime	DATETIME	YES	NO	The time when a request happens.

Entity 'Friend_Response'

Description

For each user responded a friendship request from other users. There will be one record for responding each request, including accept, deny or withdraw.

Attributes

Key	Attribute	Data Type	Not Null	Unique	Description
PFK	ReponseID	INT	YES	YES	This is a foreign-key, coming from the entity Friend_Request.
	ReponseType	VARCHAR(45)	YES	NO	For each request, the receiver can choose to "accept", "deny", while the send may "withdraw" a request. These 3 actions are stored in this attribute.

Entity 'Unfriend'

Description

For the records in the "Friend_Response" entity whose "ResponseType" is "accept". There may be a related record in the entity "Unfriend" which means some of the two friends may arise a "unfriend" action. Similar to the "Friend_Request" entity, "Unfirend Entity" also has two foreign keys – "FromUserID" and "ToUserID" working as two foreign keys to "End_User".

Attributes

Key	Attribute	Data Type	Not Null	Unique	Description
PFK	UnfriendID	INT	YES	YES	A foreign-key, coming from the entity Friend_Response.
FK	FromUserID	INT	YES	NO	A foreign-key. It records who start the "unfriend" action. It is referred to the "End_User" entity, which can help find the related user information based on the ID.
FK	ToUserID	INT	YES	NO	A foreign-key. It records to whom the "unfriend" action is performed. It is referred to the "End_User" entity, which can help find the related user information based on the ID.
	UnfriendTime	DATETIME	YES	NO	The time on which a "unfriend" action happens.

Entity 'Login'

Description

Each time the user login the website, the database will record the related login data.

Attributes

Key	Attribute	Data Type	Not Null	Unique	Description
PK	LoginID	INT	YES	YES	This is the primary-key of the entity Login, generated when a login is performed.
FK	UserID	INT	YES	NO	A foreign-key, coming from the entity User, which records who perform the login action.
	Password	VARCHAR(45)	YES	NO	This holds the password corresponding the UserID.
	Useragent	TEXT	YES	NO	This holds the user information including browser, certain system details and so on.

Entity 'Click_Event'

Description

This entity represents the events when users make every click on the website at every time. All the click actions happen after users login. There are only two types, one for navigating to a new page, another for performing a function on a webpage.

Attributes

Key	Attribute	Data Type	Not Null	Unique	Description
PK	ClickID	INT	YES	YES	This is the primary-key of the entity Click_Event, generated when a new click is made.
FK	LoginID	INT	YES	NO	A foreign-key, coming from the entity Login.
	EventType	VARCHAR(100)	YES	NO	Totally two types. Every click belongs to one of the two types.
	ClickTime	DATETIME	YES	NO	The time when a click happens.

Entity ‘Navigate_Event’

Description

This entity represents the one type of the events for navigating to a new page. For each navigation, it will locate to a new address (e.g. URL).

Attributes

Key	Attribute	Data Type	Not Null	Unique	Description
PFK	NaviClickID	INT	YES	YES	This is the foreign-key coming from the entity Click_Event, generated when a user made a new navigation to a new page.
	ToAddress	TEXT	YES	NO	This holds the address of the page of every navigation.

Entity ‘Function_Event’

Description

This entity represents another one type of the events for performing a function on a webpage.

Attributes

Key	Attribute	Data Type	Not Null	Unique	Description
PFK	FuncClickID	INT	YES	YES	This is the foreign-key coming from the entity Click_Event, generated when a user performed a new function.
	FunctionName	VARCHAR(100)	YES	NO	This represents the name of each kind of function.

Entity ‘Forum’

Description

The website is made up of the forums with different topic. They are created and opened by some moderator.

Attributes

Key	Attribute	Data Type	Not Null	Unique	Description
PK	ForumID	INT	YES	YES	This is the primary key which is the representation of the forum.
FK	OpenModeratorID	INT	YES	NO	Each forum is correspondingly created by a “Moderator”. The “OpenModeratorID” is the foreign key that is related to the entity ”Moderator”.
FK	CloseModeratroID	INT	NO	NO	Each forum can also only be closed by some of the moderators. If the forum is not closed, the value of this attribute will be null.
	ForumTopic	VARCHAR(100)	YES	NO	Every forum has one topic. Different Forums may have the some topic.
	URL	TEXT	YES	YES	Different forum will have a corresponding URL to represent its address on the website.
	CreateDate	DATE	YES	NO	The date when the forum is created will be stored in this field.
	CloseDate	DATE	NO	NO	When the forum is closed, the time of closing need to be recorded.
	Description	TEXT	YES	YES	Every forum has a short one-or two-sentence description of the topic.

Entity 'Post'

Description

The Posts are created under certain Forum, which is about the topic of the Forum. Any kind of user can create a Post.

Attributes

Key	Attribute	Data Type	Not Null	Unique	Description
PK	PostID	INT	YES	YES	This is the primary key which is the representation of the post.
FK	PostUserID	INT	YES	NO	Each post is created by certain user. The PostUserID is the foreign key which relates this entity to entity "User".
	CreatTime	DATETIME	YES	NO	This attribute records the time when the Post is created.
	ForumID	INT	YES	NO	Each post should belong to some forum. The ForumID is the foreign key that relates the Entity "Post" to "Forum".
	PhotoAddress	TEXT	NO	NO	A post may contain some photo. The photo is stored in the system file server. This attribute only store the path of the photo.
	LinkToWeb	TEXT	NO	NO	Any post may include one link to webpage.
	LinkToYoutube	TEXT	NO	NO	Any post may include one link to Youtube.
	OtherContent	TEXT	NO	NO	Any post may also include some other text content besides the PhotoAddress, link to web and link to Youtube.

Entity 'Comment'

Description

The Comments are created by the User under the Post. Any kind of the users can write some comment about the post.

Attributes

Key	Attribute	Data Type	Not Null	Unique	Description
PK	CommentID	INT	YES	YES	This is the primary key which is the representation of the comment.
	CommentUserID	INT	YES	NO	Each comment is created by certain user. The CommentID is the foreign key which relates the entity to the entity "User".
FK	PostID	INT	YES	NO	Any comment belongs to some Post. The PostID is the foreign key that relates the comment to the post.
	CreateTime	DATETIME	YES	NO	When the comment is created, the time should be sotred .
	PhotoAddress	TEXT	NO	NO	A comment may contain some photo. The photo is stored in the system file server. This attribute only store the path of the photo.
	LinkToWeb	TEXT	NO	NO	Any comment may include one link to webpage.
	LinkToYoutube	TEXT	NO	NO	Any comment may include one link to Youtube.

	OtherContent	TEXT	NO	NO	Any comment may also include some other text content besides the PhotoAddress, link to web and link to Youtube.

Entity 'Like_Comment'

Description

This is an Associative Entity between User and Comment. It records the likes that the user gives to the comment. Each record is a like between a user and a comment. Each User can like one comment once. The primary key is combined by two foreign keys, each of which is related to each of the two associated entities. This confirm that the likes of the comment are unique.

Attributes

Key	Attribute	Data Type	Not Null	Unique	Description
PFK	LikeUserID	INT	YES	NO	The LikeUserID is the UserID of User who performs a like action.
PFK	LikeCommentID	INT	YES	NO	The LikeCommentID is the liked comment by the corresponding user.

Entity 'Like_Post'

Description

This is an Associative Entity between User and Post. It records the likes that the user gives to the comment. Each record is a like between a user and a post. Each User can like one post once. The primary key is combined by two foreign keys, each of which is related to each of the two associated entities. This confirm that the likes of the post are unique.

Attributes

Key	Attribute	Data Type	Not Null	Unique	Description
PFK	LikeUserID	INT	YES	NO	The LikeUserID is the UserID of User who performs a like action.
PFK	LikePostID	INT	YES	NO	The LikePostID is the liked comment by the corresponding user.

Entity 'Tag'

Description

The time each post or comment is created, the user is advised to select some tags to describe the post or the comment. The Tag entity records the tags that exist in the comment and post.

Attributes

Key	Attribute	Data Type	Not Null	Unique	Description
PK	TagID	INT	YES	YES	The TagID is the primary key that is the representation of the Tag.
	TagName	VARCHAR(45)	YES	YES	The TagName is the name of the Tag. They come from the keywords provided to the users when they are

Entity 'Comment_Tag'

Description

One function of the websites is to find the related comments by the given Tag. This is an Associative Entity between Tag and Comment. It records the related comments of some tag and the tags of a specific comment.

Attributes

Key	Attribute	Data Type	Not Null	Unique	Description
PFK	CommentID	INT	YES	NO	The CommentID is the foreign key that refers to the Comment entity. It can help find the related comments to a specific tag.
PFK	TagID	INT	YES	NO	The TagID is the foreign key that refers to the Tag entity. It can help find the related tags to a specific comment

Entity 'Post_Tag'

Description

One function of the websites is to find the related posts by the given Tag. This is an Associative Entity between Tag and Post. It records the related posts of some tag and the tags of a specific post.

Attributes

Key	Attribute	Data Type	Not Null	Unique	Description
PFK	PostID	INT	YES	NO	The PostID is the foreign key that refers to the Post entity. It can help find the related posts to a specific tag.
PFK	TagID	INT	YES	NO	The TagID is the foreign key that refers to the Tag entity. It can help find the related tags to a specific post.