

Layouts

1. Main Layout..... 1

Tables

kpc353_2.content [1].....2

kpc353_2.content_comment [1].....2

kpc353_2.content_group_permissions [1].....2

kpc353_2.content_link_relationship [1].....3

kpc353_2.content_member_permission [1].....3

kpc353_2.content_moderation_warning [1].....4

kpc353_2.content_public_permissions [1].....4

kpc353_2.gift_registry [1].....5

kpc353_2.gift_registry_gifts [1].....5

kpc353_2.gift_registry_ideas [1].....5

kpc353_2.gift_registry_participants [1].....6

kpc353_2.gift_registry_permissions [1].....6

kpc353_2.group_event [1].....7

kpc353_2.group_event_option_vote [1].....7

kpc353_2.group_event_options [1].....8

kpc353_2.group_members [1].....8

kpc353_2.group_vote_plebiscite [1].....9

kpc353_2.group_vote_plebiscite_results [1].....9

kpc353_2.groups [1].....9

kpc353_2.member_categories [1].....10

kpc353_2.member_category_assignments [1].....10

kpc353_2.member_messages [1].....10

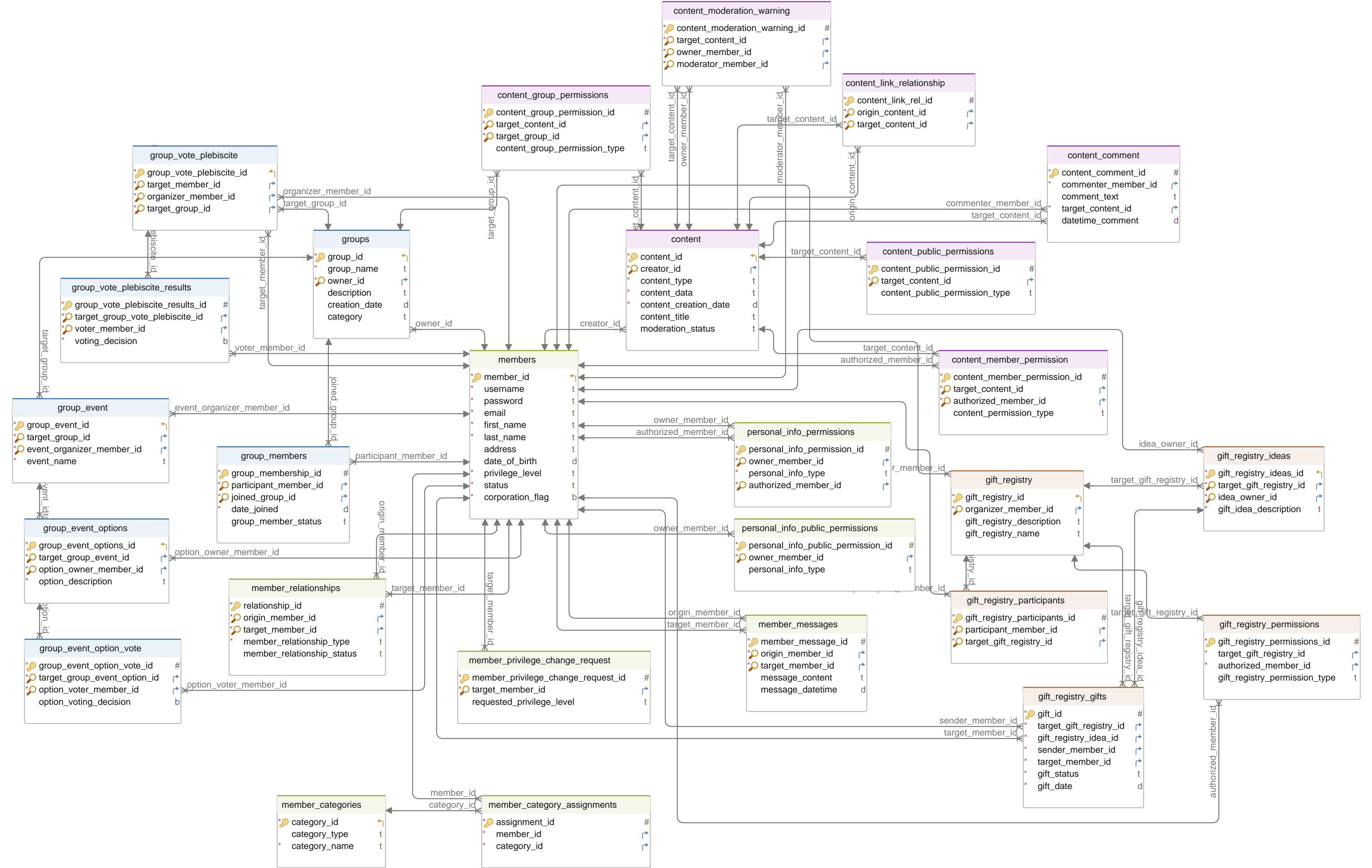
kpc353_2.member_privilege_change_request [1].....11

kpc353_2.member_relationships [1].....11

kpc353_2.members [1].....12

kpc353_2.personal_info_permissions [1].....12

kpc353_2.personal_info_public_permissions [1].....13



Main Layout

Table content			
Idx	Name	Data Type	Description
contains the content created by members			
* Pk	content_id	INT UNSIGNED AUTO_INCREMENT	
* Idx	creator_id	INT UNSIGNED	member who created this particular piece of content
*	content_type	ENUM('text','image','video')	what kind of content was produced
*	content_data	TEXT	text or URL link to the data of the content
*	content_creation_date	DATETIME DEFAULT CURRENT_TIMESTAMP	when was the content created
	content_title	VARCHAR(100)	title of the post of the content
	moderation_status	ENUM('pending', 'approved', 'rejected') DEFAULT 'pending'	status of the piece of content in terms of moderation
Indexes			
Type	Name	On	Description
Pk	pk_content_data	content_id	
	fk_content_members	creator_id	
Foreign Keys			
Type	Name	On	Description
	fk_content_members (creator_id) ref members (member_id)		

Table content_comment			
Idx	Name	Data Type	Description
table containing comments on content from members			
* Pk	content_comment_id	INT UNSIGNED AUTO_INCREMENT	
*	commenter_member_id	INT UNSIGNED	the member who made the content
	comment_text	VARCHAR(250)	the actual text of the comment on a certain piece of content
*	target_content_id	INT UNSIGNED	the content which is the target of the comment
	datetime_comment	DATETIME	the datetime of when the comment was created
Indexes			
Type	Name	On	Description
Pk	pk_content_comment	content_comment_id	
Foreign Keys			
Type	Name	On	Description
	fk_content_comment_members (commenter_member_id) ref members (member_id)		
	fk_content_comment_content (target_content_id) ref content (content_id)		

Table content_group_permissions			
Idx	Name	Data Type	Description
table describing which groups can access to what content			
* Pk	content_group_permission_id	INT UNSIGNED AUTO_INCREMENT	
* Idx	target_content_id	INT UNSIGNED	the specific piece of conent, defined by content_id on which a particular group has certain permission
* Idx	target_group_id	INT UNSIGNED	the particular group which has a certain permission on a specific content

Table content_group_permissions

	content_group_permission_type	ENUM('read','comment','share','link','edit')	the type of permission that a particular group has on a certain piece of content
Indexes			
Type	Name	On	Description
Pk	pk_content_group_permissions	content_group_permission_id	
	fk_content_group_permissions_content	target_content_id	
	fk_content_group_permissions_groups	target_group_id	
Foreign Keys			
Type	Name	On	Description
	fk_content_group_permissions_content	(target_content_id) ref content (content_id)	
	fk_content_group_permissions_groups	(target_group_id) ref groups (group_id)	

Table content_link_relationship

Idx	Name	Data Type	Description
Describes the way that content s linked between each other such as a comment to a post			
* Pk	content_link_rel_id	INT UNSIGNED AUTO_INCREMENT	
* Idx	origin_content_id	INT UNSIGNED	ContentID of the piece of content to which another piece of content is linked
* Idx	target_content_id	INT UNSIGNED	Identifies the piece of content that is linked to the origin contentID
Indexes			
Type	Name	On	Description
Pk	pk_content_link_relationship	content_link_rel_id	
	fk_content_link_relationship_content	origin_content_id	
	fk_content_link_relationship_content_0	target_content_id	
Foreign Keys			
Type	Name	On	Description
	fk_content_link_relationship_content_1	(origin_content_id) ref content (content_id)	
	fk_content_link_relationship_content_2	(target_content_id) ref content (content_id)	

Table content_member_permission

Idx	Name	Data Type	Description
allows for setting the granular permissions to a particular piece of content, targeted at a particular member			
* Pk	content_member_permission_id	INT UNSIGNED AUTO_INCREMENT	
* Idx	target_content_id	INT UNSIGNED	the specific piece of content, defined by content_id on which a particular member has a certain permission
* Idx	authorized_member_id	INT UNSIGNED	which member_id has a specific permission to do something with a specific content_id
			authorized_member_id = 1 is "private" system member authorized_member_id = 2 is "public" system member
	content_permission_type	ENUM('read','edit','comment','share','link')	the type of permission that the authorized_member_id has on this particular piece of content can be 'read','edit','comment','share','link'
Indexes			
Type	Name	On	Description

Table content_member_permission

Pk	pk_content_permissions	content_member_permission_id
	fk_content_permissions_content	target_content_id
	fk_content_permissions_members	authorized_member_id

Foreign Keys

Type	Name	On	Description
	fk_content_permissions_members	(authorized_member_id) ref members (member_id)	
	fk_content_member_permission_content	(target_content_id) ref content (content_id)	

Table content_moderation_warning

Idx	Name	Data Type	Description
	Contains the moderation warning for content posted by a member (if the content was "flagged" by a moderator)		
* Pk	content_moderation_warning_id	INT UNSIGNED AUTO_INCREMENT	
* Idx	target_content_id	INT UNSIGNED	defines the particular piece of content that has been flagged by a moderator
* Idx	owner_member_id	INT UNSIGNED	member responsible for posting a flagged content
* Idx	moderator_member_id	INT UNSIGNED	ID of a moderator who flagged the particular piece of content

Indexes

Type	Name	On	Description
Pk	pk_content_moderation_warning	content_moderation_warning_id	
	fk_content_moderation_warning_content	target_content_id	
	fk_content_moderation_warning_content_0	owner_member_id	
	fk_content_moderation_warning_members	moderator_member_id	

Foreign Keys

Type	Name	On	Description
	fk_content_moderation_warning_content	(target_content_id) ref content (content_id)	
	fk_content_moderation_warning_content_0	(owner_member_id) ref content (creator_id)	
	fk_content_moderation_warning_members	(moderator_member_id) ref members (member_id)	

Table content_public_permissions

Idx	Name	Data Type	Description
	Defines which content is public		
* Pk	content_public_permission_id	INT UNSIGNED AUTO_INCREMENT	The synthetic PK (surrogate key)
* Idx	target_content_id	INT UNSIGNED	the specific piece of content, defined by content_id on which the public has certain permission
	content_public_permission_type	ENUM('read','comment','share','link')	the type of permission that the public has on this particular piece of content can be 'read','comment','share','link'

Indexes

Type	Name	On	Description
Pk	pk_content_public	content_public_permission_id	

Table content_public_permissions

fk_content_public_permissions_c
content target_content_id

Foreign Keys

Type	Name	On	Description
	fk_content_public_permissions_content (target_content_id) ref content (content_id)		

Table gift_registry

Idx	Name	Data Type	Description
-----	------	-----------	-------------

Describes gift registry entity that different members can attach gift ideas to.

* Pk	gift_registry_id	INT UNSIGNED AUTO_INCREMENT	
* Idx	organizer_member_id	INT UNSIGNED	ID of a particular gift registry organizer
	gift_registry_description	VARCHAR(100)	Description for the particular gift registry
	gift_registry_name	VARCHAR(100)	name of the particular registry

Indexes

Type	Name	On	Description
Pk	pk_gift_registry	gift_registry_id	
	fk_gift_registry_members	organizer_member_id	

Foreign Keys

Type	Name	On	Description
	fk_gift_registry_members (organizer_member_id) members (member_id)		

Table gift_registry_gifts

Idx	Name	Data Type	Description
-----	------	-----------	-------------

contains gifts sent from gift registry

* Pk	gift_id	INT UNSIGNED AUTO_INCREMENT	
*	target_gift_registry_id	INT UNSIGNED	the gift registry from which the gift is taken
*	gift_registry_idea_id	INT UNSIGNED	target gift idea from a specific gift registry
*	sender_member_id	INT UNSIGNED	the member who is sending the gift
*	target_member_id	INT UNSIGNED	the member who will receive the gift
*	gift_status	ENUM('pending','sent','received') DEFAULT 'pending'	the status of the gift transaction
*	gift_date	DATETIME DEFAULT CURRENT_TIMESTAMP	the datetime when the gift was sent out

Indexes

Type	Name	On	Description
Pk	pk_gift_registry_gifts	gift_id	

Foreign Keys

Type	Name	On	Description
	fk_gift_registry_gifts_gift_registry (target_gift_registry_id) gift_registry (gift_registry_id)		
	fk_gift_registry_gifts_gift_registry_ideas (gift_registry_idea_id) ref gift_registry_ideas (gift_registry_ideas_id)		
	fk_gift_registry_gifts_members (sender_member_id) members (member_id)		
	fk_gift_registry_gifts_members_0 (target_member_id) members (member_id)		

Table gift_registry_ideas

Idx	Name	Data Type	Description
describes gift ideas for a particular registry			
* Pk	gift_registry_ideas_id	INT UNSIGNED AUTO_INCREMENT	
* Idx	target_gift_registry_id	INT UNSIGNED	describes which particular gift registry a gift idea applies to
* Idx	idea_owner_id	INT UNSIGNED	ID of a member proposing a particular gift
*	gift_idea_description	VARCHAR(200)	description of a proposed gift
Indexes			
Type	Name	On	Description
Pk	pk_gift_registry_ideas	gift_registry_ideas_id	
	fk_gift_registry_ideas_gift_registry	target_gift_registry_id	
	fk_gift_registry_ideas_gift_registry_participants	idea_owner_id	
Foreign Keys			
Type	Name	On	Description
	fk_gift_registry_ideas_gift_registry_0 (target_gift_registry_id) ref gift_registry (gift_registry_id)		
	fk_gift_registry_ideas_members (idea_owner_id) ref members (member_id)		

Table gift_registry_participants

Idx	Name	Data Type	Description
Contains the participants of a particular gift registry			
* Pk	gift_registry_participants_id	INT UNSIGNED AUTO_INCREMENT	
* Idx	participant_member_id	INT UNSIGNED	ID of a participant of the registry
* Idx	target_gift_registry_id	INT UNSIGNED	describes a particular gift registry that a member is part of
Indexes			
Type	Name	On	Description
Pk	pk_gift_registry_participants	gift_registry_participants_id	
	fk_gift_registry_participants_gift_registry	target_gift_registry_id	
	fk_gift_registry_participants_members	participant_member_id	
Foreign Keys			
Type	Name	On	Description
	fk_gift_registry_participants_members (participant_member_id) ref members (member_id)		
	fk_gift_registry_participants_gift_registry (target_gift_registry_id) ref gift_registry (gift_registry_id)		

Table gift_registry_permissions

Idx	Name	Data Type	Description
contains the permissions that members have on a gift registry			
* Pk	gift_registry_permissions_id	INT UNSIGNED AUTO_INCREMENT	
*	target_gift_registry_id	INT UNSIGNED	The target ID of the gift registry on which a particular members permission is defined
*	authorized_member_id	INT UNSIGNED	the ID of the member that has a certain permission on a certain gift registry

Table gift_registry_permissions

gift_registry_permission_type	ENUM('view', 'edit', 'add-item')	the type of permission that a certain member has on a certain gift registry can be 'view', 'edit', 'add-item'
-------------------------------	----------------------------------	--

Indexes

Type	Name	On	Description
Pk	pk_gift_registry_permissions	gift_registry_permissions_id	

Foreign Keys

Type	Name	On	Description
	fk_gift_registry_permissions_gift_registry (target_gift_registry_id)	ref gift_registry (gift_registry_id)	
	fk_gift_registry_permissions_members (authorized_member_id)	ref members (member_id)	

Table group_event

Idx	Name	Data Type	Description
contains the events organized for particular groups			
* Pk	group_event_id	INT UNSIGNED AUTO_INCREMENT	
* Idx	target_group_id	INT UNSIGNED	Group for which a particular event is organized
* Idx	event_organizer_member_id	INT UNSIGNED	ID of a member who is organizing a particular event
*	event_name	VARCHAR(100)	Name of the particular event being organized

Indexes

Type	Name	On	Description
Pk	pk_group_event	group_event_id	
	fk_group_event_groups	target_group_id	
	fk_group_event_members	event_organizer_member_id	

Foreign Keys

Type	Name	On	Description
	fk_group_event_groups (target_group_id)	ref groups (group_id)	
	fk_group_event_members (event_organizer_member_id)	ref members (member_id)	

Table group_event_option_vote

Idx	Name	Data Type	Description
contains the voting results of a particular member for a particular event option proposed			
* Pk	group_event_option_vote_id	INT UNSIGNED AUTO_INCREMENT	
* Idx	target_group_event_option_id	INT UNSIGNED	ID of the option of proposed date/time/place for a particular event
* Idx	option_voter_member_id	INT UNSIGNED	ID of the member who is voting on a particular option
	option_voting_decision	BOOLEAN	describes the decision of a particular member regarding the proposed date/time/place for an event (true = support for the option)

Indexes

Type	Name	On	Description
Pk	pk_group_event_option_vote	group_event_option_vote_id	
	fk_group_event_option_vote_group_event_options	target_group_event_option_id	
	fk_group_event_option_vote_members	option_voter_member_id	

Foreign Keys

Table group_event_option_vote

Type	Name	On	Description
------	------	----	-------------

fk_group_event_option_vote_group_event_options (target_group_event_option_id) ref group_event_options (group_event_options_id)

fk_group_event_option_vote_members (option_voter_member_id) ref members (member_id)

Table group_event_options

Idx	Name	Data Type	Description
-----	------	-----------	-------------

describes the proposed options for a particular group event

* Pk	group_event_options_id	INT UNSIGNED AUTO_INCREMENT	
------	------------------------	-----------------------------	--

* Idx	target_group_event_id	INT UNSIGNED	describes a particular group event ID on which a particular time/place/date option applies
-------	-----------------------	--------------	--

* Idx	option_owner_member_id	INT UNSIGNED	ID of the member proposing a particular option of date/time/place for the event
-------	------------------------	--------------	---

*	option_description	VARCHAR(100)	describes the details of the proposed option for date/time/place Ex: "Alaska, 2025-01-01, 07:00"
---	--------------------	--------------	---

Indexes

Type	Name	On	Description
------	------	----	-------------

Pk	pk_group_event_options	group_event_options_id	
----	------------------------	------------------------	--

fk_group_event_options_group_event target_group_event_id

fk_group_event_options_member option_owner_member_id

Foreign Keys

Type	Name	On	Description
------	------	----	-------------

fk_group_event_options_group_event (target_group_event_id) ref group_event (group_event_id)

fk_group_event_options_members (option_owner_member_id) ref members (member_id)

Table group_members

Idx	Name	Data Type	Description
-----	------	-----------	-------------

Mapping between members and groups, each row telling us which member belongs to which group

* Pk	group_membership_id	INT UNSIGNED AUTO_INCREMENT	
------	---------------------	-----------------------------	--

* Idx	participant_member_id	INT UNSIGNED	the member_id of the participant of this group
-------	-----------------------	--------------	--

* Idx	joined_group_id	INT UNSIGNED	the ID of a particular group, that a particular member has joined
-------	-----------------	--------------	---

*	date_joined	DATE DEFAULT CURRENT_DATE	the date when a particular member has joined a particular group
---	-------------	---------------------------	---

	group_member_status	ENUM('member','admin','requested','ban') DEFAULT 'member'	the status of a particular member in a particular group can be either 'admin', 'member', 'requested', 'ban'
--	---------------------	---	---

Indexes

Type	Name	On	Description
------	------	----	-------------

Pk	pk_group_members	group_membership_id	
----	------------------	---------------------	--

fk_group_members_members participant_member_id

fk_group_members_groups joined_group_id

Foreign Keys

Type	Name	On	Description
------	------	----	-------------

fk_group_members_members (participant_member_id) ref members (member_id)

fk_group_members_groups (joined_group_id) ref groups (group_id)

Table group_vote_plebiscite

Idx	Name	Data Type	Description
-----	------	-----------	-------------

Contains the plebiscite organized to oust a non-person (corporate) member.

* Pk	group_vote_plebiscite_id	INT UNSIGNED AUTO_INCREMENT	
* Idx	target_member_id	INT UNSIGNED	ID of a member being ousted
* Idx	organizer_member_id	INT UNSIGNED	ID of the organizer of the plebiscite
* Idx	target_group_id	INT UNSIGNED	Group from which the non-person member is being ousted.

Indexes

Type	Name	On	Description
------	------	----	-------------

Pk	pk_group_vote_plebiscite	group_vote_plebiscite_id	
	fk_group_vote_plebiscite_groups	target_group_id	
	fk_group_vote_plebiscite_members	target_member_id	
	fk_group_vote_plebiscite_members_0	organizer_member_id	

Foreign Keys

Type	Name	On	Description
------	------	----	-------------

	fk_group_vote_plebiscite_groups (target_group_id) ref groups (group_id)		
	fk_group_vote_plebiscite_members (target_member_id) ref members (member_id)		
	fk_group_vote_plebiscite_members_0 (organizer_member_id) ref members (member_id)		

Table group_vote_plebiscite_results

Idx	Name	Data Type	Description
-----	------	-----------	-------------

Results of the plebiscite

* Pk	group_vote_plebiscite_results_id	INT UNSIGNED AUTO_INCREMENT	
* Idx	target_group_vote_plebiscite_id	INT UNSIGNED	ID of the plebiscite.
* Idx	voter_member_id	INT UNSIGNED	ID of the voting member on this particular plebiscite
*	voting_decision	BOOLEAN	describes the decision of a particular member in the plebiscite vote (voting decision = true, meaning in favour of ousting)

Indexes

Type	Name	On	Description
------	------	----	-------------

Pk	pk_group_vote_plebiscite_results	group_vote_plebiscite_results_id	
	fk_group_vote_plebiscite_results_members	voter_member_id	
	fk_group_vote_plebiscite_results_group_vote_plebiscite	target_group_vote_plebiscite_id	

Foreign Keys

Type	Name	On	Description
------	------	----	-------------

	fk_group_vote_plebiscite_results_members (voter_member_id) ref members (member_id)		
	fk_group_vote_plebiscite_results_group_vote_plebiscite (target_group_vote_plebiscite_id) ref group_vote_plebiscite (group_vote_plebiscite_id)		

Table groups

Idx	Name	Data Type	Description
-----	------	-----------	-------------

Contains the information about the groups, such as their description, who created them, etc

* Pk	group_id	INT UNSIGNED AUTO_INCREMENT	
------	----------	--------------------------------	--

Table groups

*	group_name	VARCHAR(100)	
* Idx	owner_id	INT UNSIGNED	ID of the member who created a particular group
	description	TEXT	Description of the group, their interests, etc
	creation_date	DATE DEFAULT current_timestamp()	Date when group was created
	category	VARCHAR(100)	defines the different categories

Indexes

Type	Name	On	Description
Pk	pk_groups	group_id	
	fk_groups_members	owner_id	

Foreign Keys

Type	Name	On	Description
	fk_groups_members (owner_id) ref members (member_id)		

Table member_categories

Idx	Name	Data Type	Description
-----	------	-----------	-------------

contains categories for the interests of the members

* Pk	category_id	INT UNSIGNED AUTO_INCREMENT	
	category_type	ENUM('interest', 'age_group', 'profession', 'region')	
*	category_name	VARCHAR(100)	

Indexes

Type	Name	On	Description
Pk	pk_member_categories	category_id	

Table member_category_assignments

Idx	Name	Data Type
* Pk	assignment_id	INT UNSIGNED AUTO_INCREMENT
*	member_id	INT UNSIGNED
*	category_id	INT UNSIGNED

Indexes

Type	Name	On
Pk	pk_member_category_assignments	assignment_id

Foreign Keys

Type	Name	On
	fk_member_category_assignments_members (member_id) ref members (member_id)	
	fk_member_category_assignments_member_categories (category_id) ref member_categories (category_id)	

Table member_messages

Idx	Name	Data Type	Description
table containing the messages that members send between each others			
* Pk	member_message_id	INT UNSIGNED AUTO_INCREMENT	
* Idx	origin_member_id	INT UNSIGNED	the member_id from who the message is sent FROM
* Idx	target_member_id	INT UNSIGNED	member_id that receives the message
	message_content	TEXT	the actual content of the message
	message_datetime	DATETIME DEFAULT CURRENT_TIMESTAMP	the datetime of when the message was sent out

Table member_messages

Indexes

Type	Name	On	Description
Pk	pk_member_messages	member_message_id	
	fk_member_messages_members	origin_member_id	
	fk_member_messages_members_0	target_member_id	

Foreign Keys

Type	Name	On	Description
	fk_member_messages_members	(origin_member_id) ref members (member_id)	
	fk_member_messages_members_0	(target_member_id) ref members (member_id)	

Table member_privilege_change_request

Idx	Name	Data Type	Description
Contains the privilege change request of a member, for example - a junior to senior member			
* Pk	member_privilege_change_request_id	INT UNSIGNED AUTO_INCREMENT	
* Idx	target_member_id	INT UNSIGNED	The ID of the member who is requesting the change
	requested_privilege_level	ENUM('senior')	desired privilege level for the target_member_id

Indexes

Type	Name	On	Description
Pk	pk_member_privilege_change_request	member_privilege_change_request_id	
	fk_member_privilege_change_request_members	target_member_id	

Foreign Keys

Type	Name	On	Description
	fk_member_privilege_change_request_members	(target_member_id) ref members (member_id)	

Table member_relationships

Idx	Name	Data Type	Description
describes the relationships of the members, specifically if they are friends, family, colleagues or blocked			
* Pk	relationship_id	INT UNSIGNED AUTO_INCREMENT	
* Idx	origin_member_id	INT UNSIGNED	the member from whom the relationship originates, for example the origin_member_id has REQUESTED a target_member_id to be friends
* Idx	target_member_id	INT UNSIGNED	the member to whom the origin_member is connected
*	member_relationship_type	ENUM('friend','family','colleague','blocked')	The type of relationship can be 'friend', 'family', 'colleague' or 'blocked'
	member_relationship_status	ENUM('requested','approved','rejected','blocked')	used to represent the evolution of the relationship from, specifically from a friend/family/colleague request to an actual confirmed relationship

Indexes

Type	Name	On	Description
Pk	pk_member_relationships	relationship_id	
	fk_member_relationships_members	origin_member_id	
	fk_member_relationships_members_0	target_member_id	

Foreign Keys

Type	Name	On	Description
------	------	----	-------------

Table member_relationships

fk_member_relationships_members (origin_member_id) ref members (member_id)

fk_member_relationships_members_0 (target_member_id) ref members (member_id)

Table members

Idx	Name	Data Type	Description
contains the info for every member of COSN			
* Pk	member_id	INT UNSIGNED AUTO_INCREMENT	member_id = 1 is "private" system member member_id = 2 is "public" system member
*	username	VARCHAR(100)	public facing user's name (pseudonym)
*	password	VARCHAR(50)	password of the user
*	email	VARCHAR(50)	Mandatory (not null), for enabling the correct system functionality.
			email of the user
			Mandatory (not null), for identity verification purpose.
*	first_name	VARCHAR(100)	Mandatory (not null), for identity verification purpose.
*	last_name	VARCHAR(100)	Mandatory (not null), for identity verification purpose.
	address	VARCHAR(100)	address of the user. Mandatory (not null), for identity verification purpose.
	date_of_birth	DATE	date of birth of the user. Mandatory (not null), for identity verification purpose.
*	privilege_level	ENUM('administrator','senior','junior') DEFAULT 'junior'	privilege status of a group member (administrator, senior, junior) Mandatory (not null), for enabling the correct system functionality.
*	status	ENUM('active','inactive','suspended') DEFAULT 'active'	the 'system' status is used for internal backend representation of "public" and "private" members
*	corporation_flag	BOOLEAN DEFAULT false	Defines whether the member is a corporation (corporation_flag = true) or an actual person (corporation_flag = false).

Indexes

Type	Name	On	Description
Pk	pk_member	member_id	

Table personal_info_permissions

Idx	Name	Data Type	Description
Contains the mapping for 'member-specific' permissions (visibility) of private information.			
Allows a particular piece of personal information, for a particular member, to be rendered visible to a particular set of other members. The set of other members can be constructed by the application in the "shape" of a family, friends or groups, by simply inserting the needed member_id to the authorized_member_id.			
* Pk	personal_info_permission_id	INT UNSIGNED AUTO_INCREMENT	
* Idx	owner_member_id	INT UNSIGNED	the member_id of the owner of this personal information
*	personal_info_type	ENUM('first_name','last_name','date_of_birth','addresses','email')	
* Idx	authorized_member_id	INT UNSIGNED	the member_id that is authorized to view this particular personal information, of a particular member
			authorized_member_id = 1, it is visible to the "private" system member authorized_member_id = 2, it is visible to the "public" system member

Indexes

Type	Name	On	Description
Pk	pk_private_info_visibility	personal_info_permission_id	

Table personal_info_permissions

fk_personal_info_visibility_member			owner_member_id
fk_personal_info_visibility_member_0			authorized_member_id
Foreign Keys			
Type	Name	On	Description
	fk_personal_info_visibility_members (owner_member_id) ref members (member_id)		
	fk_personal_info_visibility_members_0 (authorized_member_id) ref members (member_id)		

Table personal_info_public_permissions

Idx	Name	Data Type
* Pk	personal_info_public_permission_id	INT UNSIGNED AUTO_INCREMENT
* Idx	owner_member_id	INT UNSIGNED
	personal_info_type	ENUM('first_name','last_name','date_of_birth','address','email')
Indexes		
Type	Name	On
Pk	pk_personal_info_public_permissions	personal_info_public_permission_id
	fk_personal_info_public_permissions_members	owner_member_id
Foreign Keys		
Type	Name	On
	fk_personal_info_public_permissions_members (owner_member_id) ref members (member_id)	