

**Cataloguing by Crowd Working Group, Grindstone Island, July 2005: "Steve" Glossary**

<b>BATCH</b>						for ingestion of data
batch_id	int			incremented		
batch_dtm	dtm			date batch was processed (completed)		
<b>IMAGES</b>						
mime_id	int	pk	req	system generated for internal		Note: we called this 'mime_id' to indicate that this system could be used for other types
filename_orig	varchar(255)			institution filename submitted		
filename	varchar(255)			system generated filename		
file_path	varchar(255)		req	path to image		
priority	int		req	institution sets priority (0 means done cataloguing)?		
image_metadata	TEXT			image specific metadata		name changed and datatype changed from varchar to TEXT
batch_id	int			process batch		
institution_id	char(4)		req	submitting institution (owner)		
object_id	varchar			institution object identifier		removed as this relationship is available through the IMAGE_OBJECT table
<b>IMAGE_OBJECT</b>						
mime_id	int		req			
object_id	int		req	associative to handle many to many		changed name of column to object_id as the system id (object_id will be the institution id)
batch_id	int		req	include batch id here to record the batch where this specific relationship was established (for instance to show where an object or image record already existed but we've created a new relationship)		
<b>OBJECTS</b>						
object_id	int	pk	req	system generated (for internal use)		when object information is sent
iobject_id	int		req	submitting institution object id		switched names setting object_id to the system id and iobject_id to the institution's object id
institution_id	char(4)		req			
batch_id	int			process batch		
acc_nbr	varchar(45)					
primary_title	varchar(255)					
creator	varchar(255)					
creation_date	varchar(255)					
materials	varchar(255)					
dimensions	varchar(255)					
credit_line	varchar(255)					
copyright	varchar(255)					
object_metadata	TEXT			additional data beyond tombstone in XML		name and datatype changed
<b>USERS</b>						
user_id	int	pk	req	system generated		
login_id	varchar(45)		req/unique	login id provided by user during account creation		
password	varchar(20)		req/hashed	password provided by user during account creation		
user_status	int		req	SET to 1 for pending, 2 when e-mail sent, 3 when activated and 4 for BAD, 0 for anonymous		
email	varchar(255)		req			

**Cataloguing by Crowd Working Group, Grindstone Island, July 2005: "Steve" Glossary**

language_id	int		req	user's preferred language	
experience	TEXT				
education	int			discrete range values in dropdown?	
age	int			dropdown or validated entry	
birthdate	date			user's birthdate	if we choose to calculate age
interests	varchar(255)			list of interests or free text?	
<b>TERM</b>					
term_id	int		req	system id	added for consistency and uniqueness
environment_id	int		req	the environment id when this term was added	
language_id	int		req	language assigned to the TERM	
term	varchar		req	the TERM, duh	
entered_dtm	datetime		req	the datetime the term was entered	
term_order	int		req	the ordinal position of the term on the page for the image	can be used to capture either the exact time if committing after each term or the session datetime
mime_id	int		req	the image to which this term was applied	for each term entered for an image in this session, the ordinal position on the interface
skipped_yn	char(1)		req/def=N	if user hits the "I got nuttin" button, set 'Y'	set to 'Y' if user cannot think of terms
category_id	int		?	if categories are used, relates term to a specific category	
<b>GROUPS</b>					
group_id	int	pk	req	system generated	to facilitate permissions; groups are additive
group_desc	varchar(255)		req	what is this group called?	
institution_id	char(4)			if group is institution-specific	
<b>USERS_GROUPS</b>					
user_id	int		req		
group_id	int		req	associative table	
<b>ENVIRONMENT</b>					
environment_id	int		req	system generated	a environment describes what the system looks like during a unit of work/time. this includes language, and the nature of work (i.e. the MODULE the user is working in) via mode and version of the mode.
environment_dtm	dtm	pk	req	timestamp the 'session'	
user_id	int	pk	req	logged user id	
mode_id	int		req	the MODULE the user is working in i.e. the way the user is interacting with the system. Examples include: 'Term Entry' mode, 'Gaming' mode, 'User Administration' mode	
search_criteria	TEXT		req	last search criteria (select statement) used in the current module.	
environment_name	varchar(255)			used when user saves a environment in progress	
language_id	int		?	if we support multiple label languages for the interface	
interface_settings	varchar(255)			record bit string or key/value pair string of interface settings during this work session	
mode_version	int		req	version of the interface for this particular mode	removed because of mode structure change where the version will be stored (see mode table)

**Cataloguing by Crowd Working Group, Grindstone Island, July 2005: "Steve" Glossary**

<b>INSTITUTION</b>					
institution_id	char(4)	pk	req	system id	
institution_name	varchar(255)		req	Name of the Institution	
preferences	varchar(255)		req	key/value pair string for preferences to turn features on or off... ex. Turn all metadata search on or off, enable/disable specific fields for searching, turn on/off display of object/image metadata, ...	
<b>USERS_INSTITUTION</b>					
					facilitates the relationship of users to institutions
user_id	int		req		
institution_id	char(4)		req		
<b>IMAGESET</b>					
					sets of images chosen by either general or admin users for grouping image for later retrieval (changed name to better reflect the entity)
set_id	int	pk	req	system generated	
user_id	int		req	user who created the set	
set_name	varchar(255)		req	saved name of set	
permissions	int		req	discreet level of access based on (ddlb) e.g. This set available to All Users, Institution Users, Institution Admin users, User who created it only, by group id???	
<b>IMAGESET_IMAGE</b>					
					stores groups of images for a user set (name change to coincide with related entity)
set_id	int		req		
mime_id	int		req		fixed name
<b>SEARCH</b>					
					search logging for metadata searches (name change)
search_id	int			system generated	consistency and uniqueness
user_id	int		req	user who initiated search	removed due to addition of environment where user id resides
search_criteria	TEXT		req	the select statement	
environment_id	int		req	relates back to user and env info	
rows_returned	int		req	number of rows returned	0 indicates null search for research purposes.
<b>COUNTER</b>					
					system table tracks id's <b>ELIMINATED this entity because using AUTOINCREMENT fields for id's (we can always put it back if needed)</b>
counter_type	varchar		req	what id are we tracking (batch,object,image,etc)	
count	int		req	what we increment to show the last used id for a given counter_type	
<b>SCRIPT</b>					
					holds code for applying to terms
script_id	int	pk	req		
script_text	text		req	the script	
script_name	varchar(255)		req	name of the script	
script_desc	text		req	what does it do	

**Cataloguing by Crowd Working Group, Grindstone Island, July 2005: "Steve" Glossary**

institution_id	char(4)		req	who owns/created this script	
<b>SCRIPT_GROUP</b>					for permissions to use scripts??
script_id	int		req		
group_id	int		req		
<b>BLOG</b>					blog style comment utility related to an image
blog_id	int		req	system generated	
environment_id	int		req		
mime_id			req		assuming this utility will be spawned from a term entry environment.
comment	text		req		
<b>CATEGORY</b>					
category_id	int	pk	req		
category_text	varchar(255)		req	various facets/categories for potential use in the interface for prompting users...	
<b>MODE</b>					
mode_id	int	pk	req	system id	
mode_type_cd	int		req	one of the several types of user interaction defined in further discussion (i.e. 1=Term Entry, 2=Gaming, 3=Admin, etc)	We can relate the values in some sort of back end code table or programmatically through the interface to set label values if they are displayed anywhere.
mode_version	decimal		req	tracking the version number (e.g. 1.0)	
active_yn	char(1)		req	whether or not this is the currently active mode	may want to implement some sort of trigger on the back end to ensure that only one mode_id of a particular mode_type_cd is active at once.
change_desc	TEXT			description of changes since last version	
<b>MODE_HISTORY</b>					for researching effectiveness of changes; mode id and version is stored with the Environment info. <b>THIS ENTITY removed because of change to MODE entity.</b>
mode_id	int		req	system id	
mode_version	int		req	version number for a specific mode	
active_version_yn	char(1)		req	is this the one we are using?	
change_desc	varchar			what has changed since last version	
<b>LANGUAGE</b>					list of available languages used in various locations
language_id	int				
language_name	varchar(45)			the name of the language	