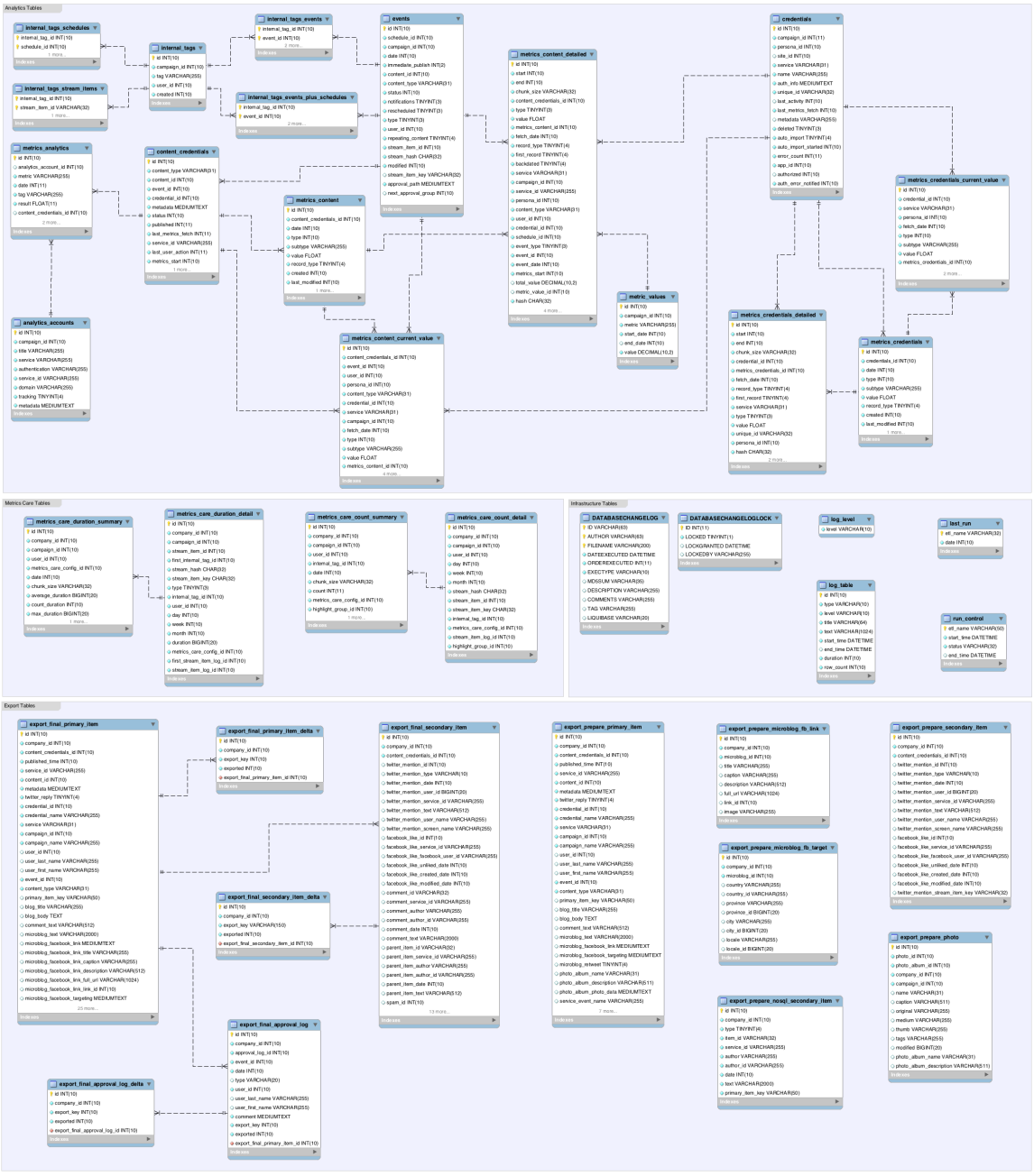
Entity Relationship Diagram - Datamart

- last_run
- log_level
- log_table
- metric_values
- metrics_analytics
- metrics_care_count_detail
- metrics_care_count_summary
- metrics_care_duration_detail
- metrics_care_duration_summary
- metrics_content
- metrics_content_current_value
- metrics_content_detailed
- metrics_credentials
- metrics_credentials_current_value
- metrics_credentials_detailed
- run_control

Description

The datamart is the schema used for reporting.

Diagram



Events

event_runAnalyticsETL

Calls the sp_runAnalyticsETL stored procedure every 2 hours.

```
CREATE EVENT event_runAnalyticsETL
    ON SCHEDULE EVERY 2 HOUR
    DO CALL sp_runAnalyticsETL();
```

event_runMetricsCareETL

Calls the event_runMetricsCareETL stored procedure every day at 2 AM Central.

```
CREATE EVENT event_runMetricsCareETL
    ON SCHEDULE EVERY 1 DAY STARTS '2011-10-22 08:00:00'
    DO CALL sp_runMetricsCareETL();
```

Stored Procedures

Logging

sp_logWrite

The sp_logWrite stored procedure is the main stored procedure that writes to the log_table. Every stored procedure in the datamart should at the very least log its beginning and end with row counts. This allows us to monitor how a stored procedure performs over time.

Parameters

Name	Data Type	Type	Description
logID	INT UNSIGNED	INOUT	The log_table.id for the row when the passed in logType is 'START' or 'END'. If the passed in logType is any other value, the logID is not updated. Note, callers need to have declared and initialized the logID before passing it in as it is passed by reference and MySQL assumes INOUT parameters have been allocated prior to the stored procedure executing.

logType	VARCHAR(10)	IN	The logType parameter is used to determine when the procedure is being called at the beginning or end of the calling stored procedure. When logType = 'START' a record i written to the log_table and the logID is updated and returned. When logType = 'END', first the logID is tested to make it is not null or zero and if so and attempt is made to look up the logID by looking for the last row that has the same logTitle and a type of 'START' and a null end_time. Next the stored procedure updates the start record setting the type to 'END', calculating the end_time and duration and updating the rest of the fields. If the passed in logType is anything other than 'START' or 'END' then a new record is simply inserted into the log_table.

logLevel	VARCHAR(10)	IN	Much like log4j, there are 5 valid logLevels: • FATAL - The FATAL level indicates a very severe error that will likely result in the application aborting. • ERROR - The ERROR level indicates an error event that might still allow the application to continue running. • WARN - The WARN level indicates potentially harmful situations • INFO - (default) This level should be used for informational message that indicate the progress of the application at a coarse-grained level. Stored procedures that are invoked during an ETL process (or on a schedule) should typically use INFO. • DEBUG - the most fine grained log level. Runtime stored procedures should typically use DEBUG. The passed in logLevel is compared to the value in the log_level table. The record is only written to the log_table iff the logLevel is sufficient.
logTitle	VARCHAR(64)	IN	The logTitle parameter is written to the log_table.title field.
logText	VARCHAR(1024)	IN	The logText parameter is written to the log_table.text field.
rowCount	INT UNSIGNED	IN	The rowCount parameter is written to the log_table.row_count field.

Table Dependencies

Read

Database	Table	Description
datamart	log_level	Each time sp_logWrite is called, the passed in logLevel is checked against the level currently configured in the log_level table. It is expected that there is only one record in the log_level table.

datamart	log_table	When the logType parameter to the sp_logWrite procedure is 'END' and the passed in logID is null, the sp_logWrite procedure attempts to lookup the last logID for the passed in logTitle where the logType is 'START' (it tries to find the start for the end).

Write

Table	Description
log_Table	The main responsibility of the sp_logWrite procedure is to write to the log_table.

Results

This stored procedure does not return a result set.

sp_logPurge

The sp_logPurge stored procedure is used to purge any records in the log_table where the log_table.start_time is before the monthsToKeep range.

Parameters

Name	Data Type	Type	Description
monthsToKeep	INT	IN	The number of months to keep in the log_table. If the passed in value is NULL or less than zero, then the value for monthsToKeep is adjusted to 6.

Table Dependencies

Read

None

Write

Table	Description
log_table	The sp_logPurge stored procedure deletes any records with a start_date greater than monthsToKeep ago.

Results

This stored procedure does not return a result set.

Analytics

sp_loadShadowTables

The sp_loadShadowTables procedure is responsible for grabbing any required tables from either the ods or the oltp replica databases. The procedure create 's_' tables (ex. when it copies the ods.metrics_content table, results are put in datamart.s_metrics_content). These tables are shadow tables. They only live for a few short minutes and if seen, they should not be queried. Shadow tables allow us to copy data in a very

consistent way while not interrupting any runtime queries. Once the shadow tables are ready, they can be atomically renamed to the live versions of the table (ex. s_metrics_content is renamed to metrics_content... for more details, see the sp_swapShadowTables procedure).

Parameters

This stored procedure does not require any parameters.

Table Dependencies

Read

Database	Table	Description
oltp replica	content_credentials	Copied from the oltp replica database and the data is used by the sp_loadMetricsContentDetailed procedure.
oltp replica	credentials	Copied from the oltp replica database and the data is used by the sp_loadMetricsContentDetailed and sp_loadMetricsCredentialsDetailed procedures.
oltp replica	events	Copied from the oltp replica database and the data is used by the sp_loadMetricsContentDetailed procedure.
oltp replica	internal_tags	Copied from the oltp replica database. There are application queries that join this table to the metrics_content_detailed table.
oltp replica	internal_tags_content	Copied from the oltp replica database. There are application queries that join this table to the metrics_content_detailed table.
oltp replica	metric_values	Copied from the oltp replica database and the data is used by the sp_loadMetricsContentDetailed procedure.
oltp replica	metrics_analytics	Copied from the oltp replica database. There are application queries that join this table to several other tables in the datamart.
oltp replica	analytics_accounts	Copied from the oltp replica database. There are application queries that join this table to several other tables in the datamart.
ods	metrics_content	Copied from the oltp replica database and the data is used by the sp_loadMetricsContentDetailed procedure.
ods	metrics_credentials	Copied from the oltp replica database and the data is used by the sp_loadMetricsCredentialsDetailed procedure.

Write

This stored procedure does not directly write to any permanent tables.

Results

This stored procedure does not return a result set.

sp_loadInternalTagsEvents

This stored procedure is responsible for maintaining the data in the internal_tags_events table. The internal_tags_events table provides an easy to query representation of the relationships between internal_tags and events. The canonical representation of these relationships is located and maintained in the internal_tags_content table. The reason the internal_tags_content table is inefficient is it requires an OR in the join criteria to get all of the relationships between internal_tags and events. This OR in the join criteria often causes queries to ignore indexes.

Parameters

This stored procedure does not require any parameters.

Table Dependencies

This store procedure does not depend directly on any permanent tables.

Results

This stored procedure does not return a result set.

sp_loadMetricsContentCurrentValue

This stored procedure is responsible for maintaining the data in the metrics_content_current_value table.

The procedure is responsible for calculating the most current value for a metric in the ods.metrics_content table (given the metric type in the oltp.metrics_config table is 'content-current-value'). The procedure does this in two passes per batch. First batches of ~batchSize are created. For each batch, we do two passes. The first is to mark the last records by date. So we sort the batch by content_credentials_id, type, subtype, and descending by date. Each time we see a new combination of content_credentials_id, type, and subtype, we mark that record as the last record. The next pass is to take all of the last records and put them in the results shadow table.

Parameters

Name	Data Type	Type	Description
batchSize	INT	IN	Used to break up the larger parts of the stored procedure into batches of content_credentials_ids to we can keep the memory usage under control as the related tables grow over time.

Table Dependencies

Read

Database	Table	Description
oltp	metrics_config	The metrics_config table houses metadata about metric type. This sp focuses on metrics types where class = 'content-current-value'.

Write

This stored procedure does not directly write to any permanent tables.

Results

This stored procedure does not return a result set.

sp_loadMetricsContentDetailed

This stored procedure is responsible for maintaining the data in the metrics_content_detailed table.

This procedure starts by setting some session variables (buffer sizes, etc) to values more appropriate for running the types of queries that are going to be run. Next the stored procedure calculates the batches based on the passed in batchSize parameter and the data in the metrics_config table. This is done by grouping the content_credential_ids in the s_metrics_content table and assigning a batch_id to each while attempting to keep each batch of metrics_content records close to the batchSize value. The reason we need to batch things up is the next few queries are expensive and we will need to work within finite buffers. Once we have the batch ids we simply start a loop and run through all the batches. Within the loop we join all the references tables and materialize the result. Next we take that result and join it to the s_metrics_content table while calculating the values for duplicate and first_record. The final step within the loop is to take those results and calculate the start and end dates for days, weeks, and months. The next big section of the procedure involves calculating metric_values. There are two types of metric values, specific and synthetic. Specific metric values are values set for a specific metric type. Synthetic metric values are values set for a list of metric types. (Specific overrides synthetic). During the metric_values calculation, both sets of values are calculated and added to the results. The next section of the stored procedure builds the s_metrics_content_detailed table. This shadow table houses the final results. This section is where we add the processing for handling chunk_size. Finally the procedure puts all fo the session variable back to their former values.

Parameters

Name	Data Type	Type	Description
batchSize	INT	IN	Used to break up the larger parts of the stored procedure into batches of content_credentials_ids to we can keep the memory usage under control as the related tables grow over time.

Table Dependencies

Read

Database	Table	Description
oltp	metrics_config	The metrics_config table houses metadata about metric type. This sp focuses on metrics types where class = 'content'.

Write

This stored procedure does not directly write to any permanent tables.

Results

This stored procedure does not return a result set.

sp_loadMetricsCredentialsCurrentValue

This stored procedure is responsible for maintaining the data in the metrics_credentials_current_value table.

The procedure is responsible for calculating the most current value for a metric in the ods.metrics_credentials table (given the metric type in the oltp.metrics_config table is 'credential-current-value'). The procedure does this in two passes per batch. First batches of ~batchSize are created. For each batch, we do two passes. The first is to mark the last records by date. So we sort the batch by credential_id, type, subtype, and descending by date. Each time we see a new combination of credential_id, type, and subtype, we mark that record as the last record. The next pass is to take all of the last records and put them in the results shadow table.

Parameters

Name	Data Type	Type	Description

batchSize	INT	IN	Used to break up the larger parts of the stored procedure into batches of credential_ids to we can keep the memory usage under control as the related tables grow over time.

Table Dependencies

Read

Database	Table	Description
oltp	metrics_config	The metrics_config table houses metadata about metric type. This sp focuses on metrics types where class = 'credential-current-value'.

Write

This stored procedure does not directly write to any permanent tables.

Results

This stored procedure does not return a result set.

sp_loadMetricsCredentialsDetailed

This stored procedure is responsible for maintaining the data in the metrics_credentials_detailed table.

This procedure starts by setting some session variables (buffer sizes, etc) to values more appropriate for running the types of queries that are going to be run. Next the stored procedure calculates the batches based on the passed in batchSize parameter and the data in the metrics_config table. This is done by grouping the credential_ids in the s_metrics_credentials table and assigning a batch_id to each while attempting to keep each batch of metrics_credentials records close to the batchSize value. Once we have the batch ids we simply start a loop and run through all the batches. The first thing we do in the loop is join the s_metrics_credentials table with the s_credentials table to pick up the necessary reference data and the calculate values for the duplicate and first_record columns. The only other thing we do in the loop is take those results and calculate the start and end dates for days, weeks, and months. Once we are out of the loop, we are ready to build the s_metrics_credentials_detailed shadow table to house the final results. This is the section where we handle the processing for chunk_size. Finally the procedure puts all fo the session variable back to their former values.

Parameters

Name	Data Type	Type	Description
batchSize	INT	IN	Used to break up the larger parts of the stored procedure into batches of credential_ids to we can keep the memory usage under control as the related tables grow over time.

Table Dependencies

Read

Database	Table	Description

oltp	metrics_config	The metrics_config table houses metadata about metric type. This sp focuses on metrics types where class = 'credential'.

Write

This stored procedure does not directly write to any permanent tables.

Results

This stored procedure does not return a result set.

sp_swapShadowTables

The sp_swapShadowTables procedure is responsible for swapping the shadow tables into the permanent table. The procedure has 3 main responsibilities:

1. drop any leftover "prior_" tables.
2. rename the shadow tables and the live tables in one atomic statement. We do this by taking all of the live tables and renaming them to prior_<table name> and renaming the s_<table name> to <table name>.
3. drop the "prior_" tables.

Parameters

This stored procedure does not require any parameters.

Table Dependencies

Read

This stored procedure does not read from any permanent tables.

Write

Table	Description
content_credentials	Renames the s_content_credentials to content_credentials.
credentials	Renames the s_credentials to credentials.
events	Renames the s_events to events.
internal_tags	Renames the s_internal_tags to internal_tags.
internal_tags_content	Renames the s_internal_tags_content to internal_tags_content.
metric_values	Renames the s_metric_values to metric_values.
metrics_content	Renames the s_metrics_content to metrics_content.
metrics_content_detailed	Renames the s_metrics_content_detailed to metrics_content_detailed.
metrics_credentials	Renames the s_metrics_credentials to metrics_credentials.
metrics_credentials_detailed	Renames the s_metrics_credentials_detailed to metrics_credentials_detailed.
metrics_analytics	Renames the s_metrics_analytics to metrics_analytics.
analytics_accounts	Renames the s_analytics_accounts to analytics_accounts.
internal_tags_events	Renames the s_internal_tags_events to internal_tags_events.

last_run	Updates the data value to the current unix_timestamp where the etl_name is Analytics_ETL.

Results

This stored procedure does not return a result set.

sp_runAnalyticsETL

The sp_runAnalyticsETL the main stored procedure that wraps up the entire ETL job. Here is the body of the stored procedure:

```
CALL sp_loadShadowTables();
CALL sp_loadInternalTagsEvents();
CALL sp_loadMetricsCredentialsDetailed(4000000);
CALL sp_loadMetricsContentDetailed(4000000);
CALL sp_loadMetricsCredentialsCurrentValue(4000000);
CALL sp_loadMetricsContentCurrentValue(4000000);
CALL sp_swapShadowTables();
CALL sp_logPurge(6);
```

The first step in the ETL (sp_loadShadowTables) loads all necessary data from the oltp replica and ods databases. This helps ensure the data we copy over is consistent. For example we do not want the application to know about records that were added to the metrics_content table since the last time we ran the ETL. The stored procedure creates shadow tables. Shadow tables are dynamically created, physical tables. Each shadow table name is prepended with an "s_". These tables only exist temporarily but cannot be implemented as actual temporary tables as they need to be passed to the next stored procedures.

The next step in the ETL creates a table called internal_tags_events. The problem this solves is when joining by event_id to get the internal_tag_id. The structure of the oltp.internal_tags_content table is such that it is nearly impossible to get an efficient query/index combination in place. So the internal_tags_events table is maintained (regenerated) with each ETL run to allow better query/index combinations

The next step in the ETL (sp_loadMetricsCredentialsDetailed) is responsible for creating the metrics_credentials_detailed data. The stored procedure uses the shadow tables created in the prior step to create the s_metrics_credentials_detailed shadow table.

The next step in the ETL (sp_loadMetricsContentDetailed) is very similar to the last except it is working on the content metrics. So it uses the shadow tables to create s_metrics_content_detailed.

The two step handle the current value metrics. They are responsible for maintaining the data in the metrics_credentials_current_value and metrics_content_current_value tables.

The next step in the ETL (sp_swapShadowTables) is responsible for changing the shadow tables into live tables. The procedure does the rename atomically.

The final step in the ETL is to purge the log_table if necessary.

The stored procedure also supports run control. This means the ETL cannot be run more than once concurrently (similar to sp_runMetricsCareETL).

Parameters

This stored procedure does not require any parameters.

Table Dependencies

This stored procedure does not directly depend on any tables.

Results

This stored procedure does not return a result set.

Export

sp_prepareCompanyExport

The sp_prepareCompanyExport procedure is responsible for copying exportable data out of the oltp replica database and into the datamart database. To do this the procedure stores its results in the export_prepare_primary_item and export_prepare_secondary_item tables. Finally the procedure is responsible for returning a result set to the Java client indicating which items may need to be pulled from the NoSQL store (Cassandra).

Parameters

Name	Data Type	Type	Description
companyID	int unsigned	IN	The Spredfast internal id of the company being exported.

Table Dependencies

Read

Database	Table	Description
oltp replica	content_credentials	Used with joins while populating the export_prepare_primary_item and export_prepare_secondary_item tables.
oltp replica	events	Used with joins while populating the export_prepare_primary_item and export_prepare_secondary_item tables.
oltp replica	credentials	Used with joins while populating the export_prepare_primary_item and export_prepare_secondary_item tables.
oltp replica	users	Used with joins while populating the export_prepare_primary_item table.
oltp replica	blogs	Used with joins while populating the export_prepare_primary_item table.
oltp replica	comments	Used with joins while populating the export_prepare_primary_item table.
oltp replica	microblogs	Used with joins while populating the export_prepare_primary_item table.
oltp replica	photo_albums	Used with joins while populating the export_prepare_primary_item table.
oltp replica	service_events	Used with joins while populating the export_prepare_primary_item table.
oltp replica	videos	Used with joins while populating the export_prepare_primary_item table.
oltp replica	campaigns	Used with joins while populating the export_prepare_secondary_item table.
oltp replica	twitter_mentions	Used with joins while populating the export_prepare_secondary_item table.
oltp replica	facebook_likes	Used with joins while populating the export_prepare_secondary_item table.

oltp replica	streams	Used to get the stream.hash column on the final result set for Facebook commentables.

Write

Table	Description
export_prepare_primary_item	One of the two main results tables. The procedure starts by deleting anything in the table for the current companyID.
export_prepare_secondary_item	One of the two main results tables. The procedure starts by deleting anything in the table for the current companyID.
export_prepare_microblog_fb_link	The procedure deletes anything in the table for the current companyID.
export_prepare_microblog_fb_target	The procedure deletes anything in the table for the current companyID.
export_prepare_photo	The procedure deletes anything in the table for the current companyID.
export_prepare_nosql_secondary_item	The procedure deletes anything in the table for the current companyID.

Results

Column	Description
primary_item_key	The export_prepare_primary_item.primary_item_key field.
stream_hash	The streams.hash column. This field is NULL when primary_item_type <> 0.
campaign_id	The export_prepare_primary_item.campaign_id field.
primary_item_type	0 iff the record related to a Facebook commentable item, 1 otherwise.
content_credentials	The max content credentials id when grouping by primary_item_key, stream_hash, and campaign_id. The Java client needs to pass a valid content_credentials_id to the Spredfast app in order to properly populate and pull from the NoSQL store.

sp_finishCompanyExport

The sp_finishCompanyExport is responsible for finishing the database work for exporting company data. The procedure takes the stored output from the prepare stored procedure and any data the Java client has persisted and puts the data into it's final format for the export. The procedure is also responsible for calculating and storing deltas. Finally, the procedure is responsible for returning exactly one record containing the current unix timestamp.

Parameters

Name	Data Type	Type	Description
companyID	int unsigned	IN	The Spredfast internal id of the company being exported.

Table Dependencies

Read

Database	Table	Description
datamart	export_prepare_primary_item	Used in several joins to populate the export_final_primary_item table.
datamart	export_prepare_microblog_fb_link	Used in a join to populate the export_final_primary_item table.
datamart	export_prepare_microblog_fb_target	Used in a join to populate the export_final_primary_item table.
datamart	export_prepare_photo	Used in a join to populate the export_final_primary_item table.
datamart	export_prepare_secondary_item	Used in several joins to populate the export_final_secondary_item table.
oltp replica	spam	Used to determine if any secondary items have been marked as spam or marked as deleted.
datamart	export_final_secondary_item_delta	Used to determine which secondary items have already been exported.
datamart	export_final_primary_item_delta	Used to determine which primary items have already been exported.

Write

Table	Description
export_final_primary_item	One of the 4 main results tables. The stores the final representation of the primary items.
export_final_secondary_item	One of the 4 main results tables. The stores the final representation of the secondary items.
export_final_primary_item_delta	One of the 4 main results tables. The stores when the primary item was last exported.
export_final_secondary_item_delta	One of the 4 main results tables. The stores when the secondary item was last exported.
export_prepare_primary_item	The procedure deletes anything in the table for the current companyID.
export_prepare_secondary_item	The procedure deletes anything in the table for the current companyID.
export_prepare_microblog_fb_link	The procedure deletes anything in the table for the current companyID.
export_prepare_microblog_fb_target	The procedure deletes anything in the table for the current companyID.
export_prepare_photo	The procedure deletes anything in the table for the current companyID.
export_prepare_nosql_secondary_item	The procedure deletes anything in the table for the current companyID.

Results

Column	Description

exportTimestamp	The unix timestamp of when the sp_finishCompanyExport was run. This value is used to name the final files and is also the value used in the final and delta tables.

sp_getFinalApprovalLog

This procedure is called by the Java client when it is time to print the final CSV.

TODO

sp_getFinalPrimaryItems

This procedure is called by the Java client when it is time to print the final CSV.

Parameters

Name	Data Type	Type	Description
companyID	int unsigned	IN	The Spredfast internal id of the company being exported.
exportedTimestamp	int unsigned	IN	The timestamp that was returned by the sp_finishCompanyExport procedure.

Table Dependencies

Read

Database	Table	Description
datamart	export_final_primary_item	The bulk of the work has already been done, now we just have to do a few minor tweaks to the data.

Write

The procedure does not write to any tables.

Results

Column	Description
content_credentials_id	export_final_primary_item.content_credentials_id
published_time	export_final_primary_item.published_time
service_id	export_final_primary_item.service_id
content_id	export_final_primary_item.content_id
credential_id	export_final_primary_item.credential_id
credential_name	export_final_primary_item.credential_name
service	export_final_primary_item.service
campaign_id	export_final_primary_item.campaign_id
campaign_name	replace(export_final_primary_item.campaign_name, "", """)

user_id	export_final_primary_item.user_id
user_last_name	replace(export_final_primary_item.user_last_name, "", """)
user_first_name	replace(export_final_primary_item.user_first_name, "", """)
event_id	export_final_primary_item.event_id
content_type	export_final_primary_item.content_type
blog_title	replace(export_final_primary_item.blog_title, "", """)
blog_body	replace(export_final_primary_item.blog_body, "", """)
comment_text	replace(export_final_primary_item.comment_text, "", """)
microblog_text	replace(export_final_primary_item.microblog_text, "", """)
microblog_facebook_link_title	replace(export_final_primary_item.microblog_facebook_link_title, "", """)
microblog_facebook_link_caption	replace(export_final_primary_item.microblog_facebook_link_caption, "", """)
microblog_facebook_link_description	replace(export_final_primary_item.microblog_facebook_link_description, "", """)
microblog_facebook_link_full_url	replace(export_final_primary_item.microblog_facebook_link_full_url, "", """)
microblog_facebook_link_link_id	replace(export_final_primary_item.microblog_facebook_link_link_id, "", """)
microblog_facebook_targeting_country	replace(export_final_primary_item.microblog_facebook_targeting_country, "", """)
microblog_facebook_targeting_country_id	replace(export_final_primary_item.microblog_facebook_targeting_country_id, "", """)
microblog_facebook_targeting_province	replace(export_final_primary_item.microblog_facebook_targeting_province, "", """)
microblog_facebook_targeting_province_id	replace(export_final_primary_item.microblog_facebook_targeting_province_id, "", """)
microblog_facebook_targeting_city	replace(export_final_primary_item.microblog_facebook_targeting_city, "", """)
microblog_facebook_targeting_city_id	replace(export_final_primary_item.microblog_facebook_targeting_city_id, "", """)
microblog_facebook_targeting_locale	replace(export_final_primary_item.microblog_facebook_targeting_locale, "", """)
microblog_facebook_targeting_locale_id	replace(export_final_primary_item.microblog_facebook_targeting_locale_id, "", """)
photo_album_name	replace(export_final_primary_item.photo_album_name, "", """)
photo_album_description	replace(export_final_primary_item.photo_album_description, "", """)
photo_id	replace(export_final_primary_item.photo_id, "", """)
photo_file_location	replace(export_final_primary_item.photo_file_location, "", """)
photo_caption	replace(export_final_primary_item.photo_caption, "", """)

service_event_name	replace(export_final_primary_item.service_event_name, "", """")
service_event_description	replace(export_final_primary_item.service_event_description, "", """")
service_event_start_time	export_final_primary_item.service_event_start_time
service_event_end_time	export_final_primary_item.service_event_end_time
service_event_location	replace(export_final_primary_item.service_event_location, "", """")
video_title	replace(export_final_primary_item.video_title, "", """")
video_description	replace(export_final_primary_item.video_description, "", """")
video_file_location	replace(export_final_primary_item.video_file_location, "", """")
exported	export_final_primary_item.exported

sp_getFinalSecondaryItems

This procedure is called by the Java client when it is time to print the final CSV.

Parameters

Name	Data Type	Type	Description
companyID	int unsigned	IN	The Spredfast internal id of the company being exported.
exportedTimestamp	int unsigned	IN	The timestamp that was returned by the sp_finishCompanyExport procedure.

Table Dependencies

Read

Database	Table	Description
datamart	export_final_secondary_item	The bulk of the work has already been done, now we just have to do a few minor tweaks to the data.

Write

The procedure does not write to any tables.

Results

Column	Description
content_credentials_id	export_final_secondary_item.content_credentials_id
twitter_mention_id	export_final_secondary_item.twitter_mention_id
twitter_mention_type	export_final_secondary_item.twitter_mention_type
twitter_mention_date	export_final_secondary_item.twitter_mention_date
twitter_mention_user_id	replace(export_final_secondary_item.twitter_mention_user_id, "", """")
twitter_mention_service_id	replace(export_final_secondary_item.twitter_mention_service_id, "", """")

twitter_mention_text	replace(export_final_secondary_item.twitter_mention_text, "", """)
twitter_mention_user_name	replace(export_final_secondary_item.twitter_mention_user_name, "", """)
twitter_mention_screen_name	replace(export_final_secondary_item.twitter_mention_screen_name, "", """)
facebook_like_id	export_final_secondary_item.facebook_like_id
facebook_like_service_id	replace(export_final_secondary_item.facebook_like_service_id, "", """)
facebook_like_facebook_user_id	replace(export_final_secondary_item.facebook_like_facebook_user_id, "", """)
facebook_like_unliked_date	export_final_secondary_item.facebook_like_unliked_date
facebook_like_created_date	export_final_secondary_item.facebook_like_created_date
facebook_like_modified_date	export_final_secondary_item.facebook_like_modified_date
facebook_comment_id	export_final_secondary_item.facebook_comment_id
facebook_comment_service_id	replace(export_final_secondary_item.facebook_comment_service_id, "", """)
facebook_comment_author	replace(export_final_secondary_item.facebook_comment_author, "", """)
facebook_comment_author_id	replace(export_final_secondary_item.facebook_comment_author_id, "", """)
facebook_comment_date	export_final_secondary_item.facebook_comment_date
facebook_comment_text	replace(export_final_secondary_item.facebook_comment_text, "", """)
parent_item_id	export_final_secondary_item.parent_item_id
parent_item_service_id	replace(export_final_secondary_item.parent_item_service_id, "", """)
parent_item_author	replace(export_final_secondary_item.parent_item_author, "", """)
parent_item_author_id	replace(export_final_secondary_item.parent_item_author_id, "", """)
parent_item_date	export_final_secondary_item.parent_item_date
parent_item_text	replace(export_final_secondary_item.parent_item_text, "", """)
spam_id	export_final_secondary_item.spam_id
spam_date_marked	export_final_secondary_item.spam_date_marked
spam_user_id	export_final_secondary_item.spam_user_id
spam_date_deleted	export_final_secondary_item.spam_date_deleted
spam_deleted_user_id	export_final_secondary_item.spam_deleted_user_id
exported	export_final_secondary_item.exported

sp_getSerializedMicroblogData

Returns a list of any non-null php serialized arrays for microblog data.

Parameters

Name	Data Type	Type	Description
companyID	int unsigned	IN	The Spredfast internal id of the company being exported.

Table Dependencies

Read

Database	Table	Description
datamart	export_prepare_primary_item	The procedure returns and non-null microblog_facebook_link or microblog_facebook_targeting data.

Write

The procedure does not write to any tables.

Results

Column	Description
content_id	export_prepare_primary_item.content_id
microblog_facebook_link	export_prepare_primary_item.microblog_facebook_link
microblog_facebook_targeting	export_prepare_primary_item.microblog_facebook_targeting

sp_getSerializedPhotoAlbumData

Returns a list of any non-null php serialized arrays for photo album data.

Parameters

Name	Data Type	Type	Description
companyID	int unsigned	IN	The Spredfast internal id of the company being exported.

Table Dependencies

Read

Database	Table	Description
datamart	export_prepare_primary_item	The procedure returns and non-null photo_album_photo_data data.

Write

The procedure does not write to any tables.

Results

Column	Description
content_id	export_prepare_primary_item.content_id

photo_album_photo_data	export_prepare_primary_item.photo_album_photo_data

Metrics Care

sp_loadMetricsCareCountDetail

The sp_loadMetricsCareCountDetail procedure is responsible for loading the metrics_care_count_detail table. The procedure handles 3 types of care count metrics:

- user by label (type = 1) - The procedure finds any records that match the metrics_care_config.internal_tag_id.
- label by users (type = 2) - The procedure finds any records that match the metrics_care_config_users.user_id.
- label expression (type = 3) - The procedure finds any records that match the metrics_care_config.query. (Note: I'm not sure this should stay in the ETL as it allows users to directly affect the ETL. The ETL has to use a cursor to loop through the metrics_care_config records.)

The procedure always checks to make sure the metrics_care_config.status = 0;

Parameters

This stored procedure does not require any parameters.

Table Dependencies

Read

Database	Table	Description
oltp replica	metrics_care_config	Used to determine what work needs to be done
oltp replica	metrics_care_config_uses	Used to determine what work needs to be done for type = 2 metrics.

Write

This stored procedure does not write to any permanent tables (it writes to the s_metrics_care_count_detail shadow table).

Results

This stored procedure does not return a result set.

sp_loadMetricsCareCountSummary

The sp_loadMetricsCareCountSummary procedure is responsible for summarizing the data in the s_metrics_care_count_detail shadow table and storing the results in the s_metrics_care_count_summary shadow table. The procedure also handles the chunk_size's.

Parameters

This stored procedure does not require any parameters.

Table Dependencies

This stored procedure does not directly read or write to any permanent tables.

Results

This stored procedure does not return a result set.

sp_loadMetricsCareDurationDetail

The sp_loadMetricsCareDurationDetail procedure is responsible for calculating the duration based care metrics stored in the stream_item_log shadow table and persisting the results in a shadow table. The procedure is calculates the duration in seconds between the first time a specified tag is applied on a given item and the last time another specified tag is applied on a given time.

Parameters

This stored procedure does not require any parameters.

Table Dependencies

This stored procedure does not directly depend on any permanent tables.

Results

This stored procedure does not return a result set.

sp_loadMetricsCareDurationSummary

The sp_loadMetricsCareDurationSummary procedure is responsible for aggregating the data created by the sp_loadMetricsCareDurationDetail procedure and persisting the results in a shadow table.

Parameters

This stored procedure does not require any parameters.

Table Dependencies

This stored procedure does not directly depend on any permanent tables.

Results

This stored procedure does not return a result set.

sp_loadMetricsCareShadowTables

The sp_loadMetricsCareShadowTables procedure is responsible for copying any tables that need to remain consistent during and between ETL runs.

Parameters

This stored procedure does not require any parameters.

Table Dependencies

Read

Database	Table	Description
ods	stream_item_log	The stream_item_log contains the entries we need to calculate the metrics care numbers.

Write

This stored procedure does not directly write to any permanent tables.

Results

This stored procedure does not return a result set.

sp_swapMetricsCareShadowTables

The sp_swapMetricsCareShadowTables procedure is responsible for atomically swapping the precalculated shadow tables with the currently online reporting tables.

Parameters

This stored procedure does not require any parameters.

Table Dependencies

Read

This stored procedure does not directly read from any permanent tables.

Write

Table	Description
metrics_care_duration_detail	Rename the s_metrics_care_duration_detail shadow table to metrics_care_duration_detail.
metrics_care_duration_summary	Rename the s_metrics_care_duration_summary shadow table to metrics_care_duration_summary.
s_stream_item_log	The s_stream_item_log table is dropped.
last_run	Updates the date for the Metrics_Care_ETL ETL.

Results

This stored procedure does not return a result set.

sp_runMetricsCareETL

The sp_runMetricsCareETL procedure is responsible for wrapping the entire metrics care ETL. The procedure also checks and updates the run_control table to make sure this ETL cannot be run concurrently. In case the ETL is accidentally run concurrently the second instance will only log the information to the datamart.log_table. The main work of the procedure is as follows:

```
CALL sp_loadMetricsCareShadowTables();
CALL sp_loadMetricsCareCountDetail();
CALL sp_loadMetricsCareCountSummary();
CALL sp_loadMetricsCareDurationDetail();
CALL sp_loadMetricsCareDurationSummary();
CALL sp_swapMetricsCareShadowTables();
```

Parameters

This stored procedure does not require any parameters.

Table Dependencies

Read

Database	Table	Description
datamart	run_control	The status of the Metrics_Care_ETL job is checked to make sure it is not 'Running'.

Write

Table	Description
run_control	The status of the Metrics_Care_ETL job is set to 'Running' while the ETL is running. If the ETL finishes without errors, the status is set to 'Complete'.

Results

This stored procedure does not return a result set.

Tables

analytics_accounts

Column Name	Description
id	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
campaign_id	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
title	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
service	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
authentication	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
service_id	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
domain	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
tracking	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
metadata	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.

content_credentials

Column Name	Description
id	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
content_type	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
content_id	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
event_id	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.

credential_id	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
metadata	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
status	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
published	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
last_metrics_fetch	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
service_id	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
last_user_action	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
metrics_start	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.

credentials

Column Name	Description
id	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
campaign_id	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
persona_id	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
site_id	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
service	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
name	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
auth_info	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
unique_id	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
last_activity	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
last_metrics_fetch	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
metadata	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
deleted	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.

auto_import	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
auto_import_started	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
error_count	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
app_id	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
authorized	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
auth_error_notified	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.

events

Column Name	Description
id	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
schedule_id	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
campaign_id	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
date	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
immediate_publish	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
content_id	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
content_type	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
status	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
notifications	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
rescheduled	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
type	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
user_id	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
repeating_content	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
stream_item_id	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.

stream_hash	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
modified	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
stream_item_key	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
approval_path	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
next_approval_group	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.

export_final_approval_log

Column Name	Description
id	The internal Spredfast primary key for the table
company_id	The company id
approval_log_id	Pulled from the approval_log.id column
event_id	Pulled from the approval_log.event_id column
date	Pulled from the approval_log.date column
type	Pulled from the approval_log.type column. Note this field is translated from an int back into a string
user_id	Pulled from the approval_log.user_id column
user_last_name	Pulled from the users.last_name column
user_first_name	Pulled from the users.first_name column
comment	Pulled from the approval_log.comment column
export_key	The key used to uniquely identify this record to the export/delta process. This field actually contains the approval_log.id
exported	Unix timestamp when the record was first exported. This field can not be null.

export_final_approval_log_delta

Column Name	Description
id	The internal Spredfast primary key for the table
company_id	The company id
export_key	matches the export_final_approval_log.exported column
exported	Unix timestamp when the record was last exported. This field can not be null.

export_final_primary_item

Column Name	Description
id	Internal primary key. The key is not a fk to any other table, it only exists for indexing purposes.
company_id	The Spredfast internal company id. This field can not be null.
content_credentials_id	The Spredfast internal content credentials id. This field can not be null. This field may be used by itself to determine whether a record has been exported however there will be duplicate values in this table (because we had to blow out the data to handle serialized php arrays)
published_time	The unix timestamp when the content credential was published. This field can not be null.
service_id	The external id of the item (provided by the service ... Facebook, Twitter, etc. This field can not be null.
content_id	The Spredfast internal id of the content. This value must be used together with the content_type as different types of content are stored in different tables. This field can not be null.
metadata	The metadata for the content credential. This field can not be null.
twitter_reply	If after introspecting the metadata field it is determined the current item is a twitter reply then the value is 1 else it is 0. This field can not be null.
credential_id	The Spredfast internal id of the credential. This field can not be null.
credential_name	The user entered Spredfast name of the credential. This field can not be null.
service	The service for the credential (facebook, twitter). This field can not be null.
campaign_id	The Spredfast internal campaign id. This field can not be null.
campaign_name	The name of the Spredfast campaign. This field can not be null.
user_id	The internal Spredfast id of the user.
user_last_name	The Spredfast user's last name.
user_first_name	The Spredfast user's first name.
event_id	The Spredfast internal id of the event. This field can not be null.
content_type	The content type of the event. This field can not be null.
primary_item_key	The key used to look up the primary item in the NoSQL store. In the case of Facebook microblogs, photo_albums, and videos, the value is made up of an md5 hash of the service id concatenated with the unix timestamp for the next Sunday minus one second else the value is the stream item key from the event.
blog_title	If the content_type is blog, this field holds the title of the blog.
blog_body	If the content_type is blog, this field holds the body of the blog.
comment_text	If the content_type is comment, this field holds the text of the comment.
microblog_text	If the content_type is microblog, this field holds the text of the microblog.

microblog_facebook_link	The php serialized array for the Facebook link information.
microblog_facebook_link_title	The title pulled out of the microblog_facebook_link serialized php array.
microblog_facebook_link_caption	The caption pulled out of the microblog_facebook_link serialized php array.
microblog_facebook_link_description	The description pulled out of the microblog_facebook_link serialized php array.
microblog_facebook_link_full_url	The full url pulled out of the microblog_facebook_link serialized php array.
microblog_facebook_link_link_id	The link id pulled out of the microblog_facebook_link serialized php array.
microblog_facebook_targeting	The php serialized array for the Facebook targeting information.
microblog_facebook_targeting_country	The country name pulled out of the microblog_facebook_targeting serialized php array.
microblog_facebook_targeting_country_id	The country id pulled out of the microblog_facebook_targeting serialized php array.
microblog_facebook_targeting_province	The province name pulled out of the microblog_facebook_targeting serialized php array.
microblog_facebook_targeting_province_id	The province id pulled out of the microblog_facebook_targeting serialized php array.
microblog_facebook_targeting_city	The city name pulled out of the microblog_facebook_targeting serialized php array.
microblog_facebook_targeting_city_id	The city id pulled out of the microblog_facebook_targeting serialized php array.
microblog_facebook_targeting_locale	The locale name pulled out of the microblog_facebook_targeting serialized php array.
microblog_facebook_targeting_locale_id	The locale id pulled out of the microblog_facebook_targeting serialized php array.
microblog_retweet	1 iff the microblog was a retweet, 0 otherwise
photo_album_name	The name of the photo album.
photo_album_description	The description of the photo album.
photo_album_photo_data	The php serialized array for the photo data information.
photo_id	The internal Spredfast photo id pulled out of the photo_album_photo_data serialized php array.
photo_file_location	The original location of the photo pulled out of the photo_album_photo_data serialized php array.
photo_caption	The caption of the photo pulled out of the photo_album_photo_data serialized php array.
service_event_name	The name of the Facebook service event.
service_event_description	The description of the Facebook service event.
service_event_start_time	The start time of the Facebook service event.
service_event_end_time	The end time of the Facebook service event.

service_event_location	The location of the Facebook service event.
video_title	The title of the video.
video_description	The description of the video.
video_file_location	The file location of the video.
export_key	Key used to determine whether this record has been exported in it's current form. This field can not be null.
exported	Unix timestamp when the record was first exported. This field can not be null.

export_final_primary_item_delta

Column Name	Description
id	Internal primary key. The key is not a fk to any other table, it only exists for indexing purposes.
company_id	The Spredfast internal company id. This field can not be null.
export_key	For primary items, the export_key is the content_credentials_id.
exported	Unix timestamp when the record was last exported. This field can not be null.

export_final_secondary_item

Column Name	Description
id	Internal primary key. The key is not a fk to any other table, it only exists for indexing purposes.
company_id	The Spredfast internal company id. This field can not be null.
content_credentials_id	The Spredfast internal content credentials id. This field can not be null. This field can be used to join to the primary item records.
twitter_mention_id	The Spredfast internal twitter mention id. All of the remaining fields whose names are prefixed by twitter_mentions_ come from the twitter_mentions table.
twitter_mention_type	Derived string indicated whether this record is a reply or retweet.
twitter_mention_date	The unix timestamp date of the twitter mention.
twitter_mention_user_id	The internal Spredfast user id.
twitter_mention_service_id	The Twitter generated service id
twitter_mention_text	The text of the reply or retweet.
twitter_mention_user_name	The Twitter generated user id.
twitter_mention_screen_name	The Twitter screen name.
facebook_like_id	The Spredfast internal facebook_like id. All of the remaining fields whose names are prefixed by facebook_like come from the facebook_likes table.
facebook_like_service_id	The Facebook generated service id. (the id for the like)

facebook_like_facebook_user_id	The Facebook generated user id.
facebook_like_unliked_date	The unix timestamp of when the content was unliked.
facebook_like_created_date	The unix timestamp od when the content was liked.
facebook_like_modified_date	The date the facebook_like was last modified.
comment_id	The Spredfast internal stream item id. All of the remaining fields whose names are prefixed by 'comment_' come from Cassandra.
comment_service_id	The service id for the comment. (generated by the service)
comment_author	The author name for the comment (set by the service)
comment_author_id	The author id of the comment (set by the service)
comment_date	The date of the comment (set by the service)
comment_text	The text of the comment
parent_item_id	The Spredfast internal stream item id for the parent item. (a parent item is an item that was not posted by Spredfast, but a comment was posted from Spredfast. When this happens, we bring in the "parent item" so we have the context.
parent_item_service_id	The id given to the parent item by the service.
parent_item_author	The author name for the parent item (set by the service)
parent_item_author_id	The author id for the parent item (set by the service)
parent_item_date	The date of the parent item (set by the service)
parent_item_text	The text of the parent item
spam_id	The internal Spredfast id assigned to the spam record
spam_date_marked	The unix timestamp date the item was marked as spam
spam_user_id	The user id for the person that marked the item as spam
spam_date_deleted	The date the item was marked as deleted
spam_deleted_user_id	The user id for the person that marked the item as deleted
export_key	The Spredfast generated key that determines if this record has changed. This value is used to calculate deltas. Deltas allows us to not send the same exact record with the same exact state to the end users.
exported	The unix timestamp the record was exported. Each export process creates a timestamp and all records exported are marked with that timestamp.
note_id	The internal Spredfast id of the note
note_user_id	The internal Spredfast id of the user that created the note
note_user_first_name	The first name of the user that created the note
note_user_last_name	The last name of the user that created the note
note_creation_time	The unix timestamp the note was created
note_text	The text of the note

export_final_secondary_item_delta

Column Name	Description
id	Internal primary key. The key is not a fk to any other table, it only exists for indexing purposes.
company_id	The Spredfast internal company id. This field can not be null.
export_key	concat(`content_credentials_id`, ':', case when `twitter_mention_id` is null then 'NULL' else `twitter_mention_id` end, ':', case when `facebook_like_id` is null then 'NULL' else `facebook_like_id` end, ':', case when `facebook_comment_id` is null then 'NULL' else `facebook_comment_id` end, ':', case when `parent_item_id` is null then 'NULL' else `parent_item_id` end, ':', case when `spam_date_marked` is null then 'NULL' else `spam_date_marked` end, ':', case when `spam_date_deleted` is null then 'NULL' else `spam_date_deleted` end)
exported	Unix timestamp when the record was last exported. This field can not be null.

export_prepare_microblog_fb_link

This table should typically be empty. The export process uses it to house temporary data created by the java client. When the export process completes, anything for the current company is deleted. The table's main purpose is to temporarily house unserialized php array information (from the microblog.fb_link column)

Column Name	Description
id	The internal Spredfast primary key for the table.
company_id	The company id
microblog_id	The microblog id
title	The title pulled from the serialized php array stored in the microblog.fb_link column
caption	The caption pulled from the serialized php array stored in the microblog.fb_link column
description	The description pulled from the serialized php array stored in the microblog.fb_link column
full_url	The url pulled from the serialized php array stored in the microblog.fb_link column
link_id	The link id pulled from the serialized php array stored in the microblog.fb_link column
image	The image text pulled from the serialized php array stored in the microblog.fb_link column

export_prepare_microblog_fb_target

This table should typically be empty. The export process uses it to house temporary data created by the java client. When the export process

completes, anything for the current company is deleted. This table is very similar to export_prepare_microblog_fb_link.

Column Name	Description
id	The internal Spredfast primary key for the table.
company_id	The company id
microblog_id	The microblog id
country	The country pulled from the serialized php array stored in the microblog.fb_target column
country_id	The country id pulled from the serialized php array stored in the microblog.fb_target column
province	The province pulled from the serialized php array stored in the microblog.fb_target column
province_id	The province id pulled from the serialized php array stored in the microblog.fb_target column
city	The city pulled from the serialized php array stored in the microblog.fb_target column
city_id	The city id pulled from the serialized php array stored in the microblog.fb_target column
locale	The locale pulled from the serialized php array stored in the microblog.fb_target column
locale_id	The locale id pulled from the serialized php array stored in the microblog.fb_target column

export_prepare_nosql_secondary_item

This table should typically be empty. The export process uses it to house temporary data created by the java client. When the export process completes, anything for the current company is deleted. The java process queries Cassandra for this information.

Column Name	Description
id	The internal Spredfast primary key for the table
company_id	The company id
type	0 = comment; 1 = parent item
item_id	The internal Spredfast id for the item (parent item or comment) stream item key
service_id	The id of the item assigned by the service
author	The author of the item assigned by the service
author_id	The author id of the item assigned by the service
date	The date of the item assigned by the service
text	The text of the item
primary_item_key	The key of the primary item. Note in some cases this key is used to hit an index that gives a list of other keys, then those keys are used to hit another index that finally has the actual stream item keys of the secondary items.

export_prepare_photo

This table should typically be empty. The export process uses it to house temporary data created by the java client. When the export process completes, anything for the current company is deleted. This table is very similar to export_prepare_microblog_fb_link.

Column Name	Description
id	The internal Spredfast primary key for the table
photo_id	The photo id
photo_album_id	The photo album id
company_id	The company id
campaign_id	The campaign id
name	The name pulled from the serialized php array stored in the photo_albums.photo_data column
caption	The caption pulled from the serialized php array stored in the photo_albums.photo_data column
original	The original field pulled from the serialized php array stored in the photo_albums.photo_data column
medium	The medium pulled from the serialized php array stored in the photo_albums.photo_data column
thumb	The thumb field pulled from the serialized php array stored in the photo_albums.photo_data column
tags	The tags pulled from the serialized php array stored in the photo_albums.photo_data column
photo_album_name	The name of the photo album pulled from the serialized php array stored in the content_credentials.metadata column. When photos are posted to existing albums, the content_credentials table is the only place to get the photo album name and description
photo_album_description	The photo album description pulled from the serialized php array stored in the content_credentials.metadata column
modified	The modified field pulled from the serialized php array stored in the photo_albums.photo_data column

export_prepare_primary_item

This table should typically be empty. The export process uses it to house temporary data created by sql and needed by the java client. When the export process completes, anything for the current company is deleted.

Column Name	Description
id	The internal Spredfast primary key for the table
company_id	The company id
content_credentials_id	The content_credentials.id. This value should be unique in this table. The value also provides a unique key for the records in the final primary file
published_time	Pulled from the content_credentials.published column
service_id	Pulled from the content_credentials.service_id column

content_id	Pulled from the content_credentials.content_id column
metadata	Pulled from the content_credentials.metadata column
twitter_reply	Calculated value. 0 if the record is not a twitter reply, 1 otherwise
credential_id	Pulled from the credentials.id column
credential_name	Pulled from the credentials.name column
service	Pulled from the credentials.service column
campaign_id	Pulled from the campaigns.id column
campaign_name	Pulled from the campaigns.name column
user_id	Pulled from the users.id column
user_last_name	Pulled from the users.last_name column
user_first_name	Pulled from the users.first_name column
event_id	Pulled from the events.id column
content_type	Pulled from the events.content_type column
primary_item_key	Calculated value. For Facebook this value is the md5(content_credentials.service_id). For everything else the value is md5(concat('content_credentials:', content_credentials.id))
blog_title	Pulled from the blogs.title column
blog_body	Pulled from the blogs.body column
comment_text	Pulled from the comments.text column
microblog_text	Pulled from the microblogs.text column
microblog_facebook_link	Pulled from the microblogs.fb_link column
microblog_facebook_targeting	Pulled from the microblogs.fb_targeting column
microblog_retweet	Pulled from the microblogs.retweet column
photo_album_name	Pulled from the photo_albums.name column
photo_album_description	Pulled from the photo_albums.description column
photo_album_photo_data	Pulled from the photo_albums.photo_data column
service_event_name	Pulled from the service_events.name column
service_event_description	Pulled from the service_events.description column
service_event_start_time	Pulled from the service_events.start_time column
service_event_end_time	Pulled from the service_events.end_time column
service_event_location	Pulled from the service_events.location column
video_title	Pulled from the videos.title column
video_description	Pulled from the videos.description column
video_file_location	Pulled from the videos.location column

export_prepare_secondary_item

This table should typically be empty. The export process uses it to house temporary data created by sql and needed by the java client. When the export process completes, anything for the current company is deleted.

Column Name	Description
id	The internal Spredfast primary key for the table
company_id	The company id
content_credentials_id	The content_credentials.id. Note, unlike in the primary item table, this filed can have duplicates. The field does serve as a key to join back to the primary items table
twitter_mention_id	Pulled from the twitter_mentions.id column
twitter_mention_type	Calculated value. 'reply' if the record is a reply, 'retweet' if the record is a retweet, 'mention' otherwise
twitter_mention_date	Pulled from the twitter_mentions.date column
twitter_mention_user_id	Pulled from the twitter_mentions.user_id column
twitter_mention_service_id	Pulled from the serialized xml in the twitter_mentions.status column
twitter_mention_text	Pulled from the serialized xml in the twitter_mentions.status column
twitter_mention_user_name	Pulled from the serialized xml in the twitter_mentions.status column
twitter_mention_screen_name	Pulled from the serialized xml in the twitter_mentions.status column
facebook_like_id	TODO
facebook_like_service_id	TODO
facebook_like_facebook_user_id	TODO
facebook_like_unliked_date	TODO
facebook_like_created_date	TODO
facebook_like_modified_date	TODO

internal_tags

Column Name	Description
id	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
campaign_id	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
tag	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
user_id	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
created	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
company_id	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.

priority_tag	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.

internal_tags_events

Column Name	Description
internal_tag_id	A reference to the internal_tags table
event_id	A reference to the events table.

internal_tags_events_plus_schedules

This table is maintained by the Analytics ETL table and provides an easy to query way to determine which internal_tag_ids go with which event_ids. The table that provides the canonical representation of the relationship is the internal_tags_content table. However this table requires an 'OR' in the join criteria for queries (which is suboptimal as it does not particularly like to use indexes).

Column Name	Description
internal_tag_id	A reference to the internal_tags table
event_id	A reference to the events table.

internal_tags_schedules

Column Name	Description
internal_tag_id	A reference to the internal_tags table
schedule_id	A reference to the schedules table.

internal_tags_stream_items

Column Name	Description
internal_tag_id	A reference to the internal_tags table
stream_item_key	The stream_item_key from Cassandra.

last_run

Column Name	Description
etl_name	The name of the ETL (ex. Analytics_ETL). This column is also the primary key.
date	The unix timestamp for the last time the ETL ran. The value is 0 if the ETL has never run.

log_level

Column Name	Description

level	Indicates the logLevel for the sp_logWrite stored procedure. Valid values are: • FATAL - The FATAL level indicates a very severe error that will likely result in the application aborting. • ERROR - The ERROR level indicates an error event that might still allow the application to continue running. • WARN - The WARN level indicates potentially harmful situations • INFO - (default) This level should be used for informational message that indicate the progress of the application at a coarse-grained It is expected that there is exactly one record in this table that may be updated at runtime by a simple custom SQL statement.

log_table

Column Name	Description
id	The primary key.
type	values 'START' and 'END' have special behavior implemented in sp_logWrite. Any other value may be used in this field, the most common being 'INFO'.
level	'FATAL', 'ERROR', 'WARN', 'INFO', or 'DEBUG'
title	The title field is normally the name of the stored procedure that is doing the logging.
text	The log text
start_time	The db server time when the record was inserted.
end_time	The db server time when the record was ended (the sp_logWrite procedure was called with logType= 'END' for the current log_table.id). The field is defaulted to the time when the record was inserted.
duration	The difference in seconds between start_time and end_time.
row_count	The row count. This value is passed in to the sp_logWrite procedure.

metric_values

Column Name	Description
id	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
campaign_id	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
metric	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
start_date	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
end_date	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.

value	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.

metrics_analytics

Column Name	Description
id	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
analytics_account_id	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
metric	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
date	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
tag	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
result	Copied from the oltp replica database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
content_credentials_id	Convenience data calculated while copying the table over. The code uses a regular expression to determine if the tag column has the format of 'sf'+<a number> (<a number> is the content_credentials_id) and if so, parses out the content_credentials_id.

metrics_care_count_detail

Column Name	Description
id	The AUTO_INCREMENT generated primary key of the table. Note this value starts over with each run (sp_loadMetricsCareCountDetail) so you cannot count on this value being consistent between runs.
company_id	The company_id sourced from the ods.stream_item_log table.
campaign_id	The campaign_id sourced from the ods.stream_item_log table.
user_id	The user_id sourced from the ods.stream_item_log table.
day	The unix timestamp for the stream_item_log.date column with the time set to 00:00:00
week	The unix timestamp for the stream_item_log.date column with day set to the prior Sunday and the time set to 00:00:00
month	The unix timestamp for the stream_item_log.date column with day set to the 1st of the month and the time set to 00:00:00
stream_hash	The stream_hash sourced from the ods.stream_item_log table.
stream_item_id	The stream_item_id sourced from the ods.stream_item_log table.
stream_item_key	The stream_item_key sourced from the ods.stream_item_log table.
internal_tag_id	The internal_tag_id sourced from the ods.stream_item_log table.
metrics_care_config_id	The id sourced from the oltp replica.metrics_care_config table.

stream_item_log_id	The id sourced from the ods.stream_item_log table.

metrics_care_count_summary

id	The AUTO_INCREMENT generated primary key of the table. Note this value starts over with each run (sp_loadMetricsCareCountSummary) so you cannot count on this value being consistent between runs.
company_id	The company_id sourced from the datamart.metrics_care_count_detail table.
campaign_id	The campaign_id sourced from the datamart.metrics_care_count_detail table.
user_id	The user_id sourced from the datamart.metrics_care_count_detail table.
internal_tag_id	The internal_tag_id sourced from the datamart.metrics_care_count_detail table.
date	if the chunk_size = day then this column is sourced from the metrics_care_count_detail.day column if the chunk_size = week then this column is sourced from the metrics_care_count_detail.week column if the chunk_size = month then this column is sourced from the metrics_care_count_detail.month column
chunk_size	day, week, month
count	The count of the records in the metrics_care_count_detail grouped by company_id, campaign_id, user_id, internal_tag_id, metrics_care_config_id, <day, week, or month>.
metrics_care_config_id	The metrics_care_config_id sourced from the datamart.metrics_care_count_detail table.

metrics_care_duration_detail

Column Name	Description
id	Auto incremented primary key. The values of this key will change between ETL runs.
company_id	The company_id sourced from the stream_item_log table.
campaign_id	The company_id sourced from the stream_item_log table.
stream_item_id	Part of the unique key for a stream item.
first_internal_tag_id	The first_internal_tag_id configured in the metrics_care_config table.
stream_hash	Part of the unique key for a stream item.
stream_item_key	Part of the unique key for a stream item.
type	The type of care metric.
internal_tag_id	The second internal_tag_id configured in the metrics_care_config table.
user_id	The user_id sourced from the stream_item_log table.

day	The unix timestamp for the stream_item_log.date column with the time set to 00:00:00
week	The unix timestamp for the stream_item_log.date column with day set to the prior Sunday and the time set to 00:00:00
month	The unix timestamp for the stream_item_log.date column with day set to the 1st of the month and the time set to 00:00:00
duration	The difference in seconds between the date of the first first_internal_tag_id and the date of the last second internal_tag_id (in the stream_item_log_table)
metrics_care_config_id	The id from the metrics_care_config table.
first_stream_item_log_id	The id sourced from the stream_item_log table for the first_internal_tag_id.
stream_item_log_id	The id sourced from the stream_item_log table for the second internal_tag_id.

metrics_care_duration_summary

Column Name	Description
id	Auto incremented primary key. The values of this key will change between ETL runs.
company_id	The company_id sourced from the metrics_care_duration_detail table.
campaign_id	The campaign_id sourced from the metrics_care_duration_detail table.
user_id	The user_id sourced from the metrics_care_duration_detail table.
metrics_care_config_id	The id from the metrics_care_duration_detail table.
date	if the chunk_size = day then this column is sourced from the metrics_care_duration_detail.day column if the chunk_size = week then this column is sourced from the metrics_care_duration_detail.week column if the chunk_size = month then this column is sourced from the metrics_care_duration_detail.month column
chunk_size	day, week, month
average_duration	The straight avg for the duration column of the metrics_care_duration_detail table grouped by company_id, campaign_id, user_id, metrics_care_config_id , <day, week, or month>.
count_duration	The straight count of the metrics_care_duration_detail table grouped by company_id, campaign_id, user_id, metrics_care_config_id , <day, week, or month>.
max_duration	The straight max for the duration column of the metrics_care_duration_detail table grouped by company_id, campaign_id, user_id, metrics_care_config_id , <day, week, or month>.

metrics_content

Column Name	Description
id	Copied from the ods database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
content_credentials_id	Copied from the ods database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
date	Copied from the ods database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
type	Copied from the ods database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
value	Copied from the ods database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
record_type	Copied from the ods database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure. (0 = normal; 1 = correction; 2 = metric_content_deleted)
created	Copied from the ods database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure. The unix timestamp when the record was inserted.
last_modified	Copied from the ods database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure. The unix timestamp when the record was last updated.

metrics_content_current_value

Column Name	Description
id	The AUTO_INCREMENT generated primary key of the table. Note this value starts over with each run (sp_loadMetricsContentCurrentValue) so you cannot count on this value being consistent between runs.
content_credentials_id	A reference to the content_credentials table. The value is pulled from the ods.metrics_content table.
event_id	A reference to the events table. The value is pulled from the oltp.content_credentials table
user_id	A reference to the oltp.users table. The value is pulled from the events.user_id column
persona_id	A reference to the oltp.personas table. The value is pulled from the oltp.credentials table.
content_type	The content type for the content credentials. The value is pulled from the oltp.content_credentials table
credential_id	A reference to the credentials table. The value is pulled from the oltp.content_credentials table.
service	The service (ex. facebook, twitter, etc). The value is pulled from the oltp.credentials table.
campaign_id	A reference to the oltp.campaigns table. The value is pulled from the oltp.events table.

fetch_date	The unix timestamp date the ods.metrics_content record was fetched. The value is pulled from the ods.metrics_content.date column.
type	The type of metric. The value is pulled from the ods.metrics_content.type column.
subtype	The String subtype of the metric. The value is ultimately set by the service. The value si pulled from the ods.metrics_content.subtype column.
value	The value of the metric. This value is pulled from the ods.metrics_content.value column.
metrics_content_id	A reference to the ods.metrics_content table.

metrics_content_detailed

Column Name	Description
id	The AUTO_INCREMENT generated primary key of the table. Note this value starts over with each run (sp_loadMetricsContentDetailed) so you cannot count on this value being consistent between runs.
start	The unix timestamp for the start date of the record. **The time portion of the value is set to 00:00:00 US/Central timezone.**
end	The unix timestamp for the end date of the record. **The time portion of the value is set to 00:00:00 US/Central timezone.** When the record has no end date, the value is 0.
chunk_size	'day', 'week', or 'month'. The chunk_size determines the granularity of the time periods.
content_credentials_id	The reference to the content_credentials table. This is the most important reference as it is the lowest level for which these metrics are recorded.
type	The type of metric. The values are hardcoded in the php application. Types of metrics are unique given the service and the type (types are reused across services).
value	The value for the metric.
metrics_content_id	A reference back to the both the datamart.metrics_content and ods.metrics_content tables. Note the data will be consistent between this table and the datamart.metrics_content table, but it is expected that the ods.metrics_content table will continue to accumulate records and therefore have more data most of the time.
fetch_date	The fetch date for this record. This value comes from the metrics_content.date field. This field contains the actual value for the time as the application server calculated (it is not set to 00:00:00 US/Central).
record_type	This field is copied over from the metrics_content.record_type field.
first_record	This field is calculated by the sp_loadMetricsContentDetailed procedure. The value is set to 1 if this is the first record for this content_credentials_id and type combination (ordered by fetch_date). The value is 0 otherwise.

backdated	Some records need to be backdated. If first_record = 1 and event_type = 2 and metrics_start > (event_date + 1 day) then backdated = 1 else backdated = 0. Note event_type 2 is TYPE_IMPORT.
service	The value for service is pulled from the credentials table.
campaign_id	The value for campaign_id is pulled from the events table.
service_id	The value for service_id is pulled from the content_credentials table.
persona_id	The value for persona_id is pulled from the credentials table.
content_type	The value for content_type is pulled from the content_credentials table.
user_id	The value for user_id is pulled from the events table.
credential_id	The value for credential_id is pulled from the content_credentials table. This value can be used to join to the credentials table.
schedule_id	The value for schedule_id is pulled from the events table.
event_type	The value for event_type is pulled from the type column in the events table.
event_id	The value for event_id is pulled from the content_credentials table. This value can be used to join to the events table.
event_date	The value for even_date is pulled from the date column in the events table.
metrics_start	The value for metrics_start is pulled from the content_credentials table.
total_value	The value for total_value is set by the sp_loadMetricsContentDetailed procedure. The procedure uses the data in the metric_values table to find and set the total_value on any matching records.
metric_value_id	The value for metric_value_id is a reference to the metric_values table and is set iff total_value is not null.
hash	md5(concat(`chunk_size`, ':', `service`, ':', `type`))

metrics_credentials

Column Name	Description
id	Copied from the ods database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
credentials_id	Copied from the ods database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
date	Copied from the ods database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
type	Copied from the ods database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.
value	Copied from the ods database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure.

record_type	Copied from the ods database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure. (0 = normal)
created	Copied from the ods database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure. The unix timestamp when the record was inserted.
last_modified	Copied from the ods database by the sp_loadShadowTables procedure and then written by the sp_swapShadowTables procedure. The unix timestamp when the record was last updated.

metrics_credentials_current_value

Column Name	Description
id	The AUTO_INCREMENT generated primary key of the table. Note this value starts over with each run (sp_loadMetricsCredentialsCurrentValue) so you cannot count on this value being consistent between runs.
credential_id	A reference to the credentials table. The value is pulled from the ods.metrics_credentials.credential_id column.
service	The service (ex. facebook, twitter, etc). The value is pulled from the oltp.credentials table.
persona_id	A reference to the oltp.personas table. The value is pulled from the oltp.credentials table.
fetch_date	The unix timestamp date the ods.metrics_credentials record was fetched. The value is pulled from the ods.metrics_credentials.date column.
type	The type of metric. The value is pulled from the ods.metrics_credentials.type column.
subtype	The String subtype of the metric. The value is ultimately set by the service. The value si pulled from the ods.metrics_credentials.subtype column.
value	The value of the metric. This value is pulled from the ods.metrics_credentials.value column.
metrics_credentials_id	A reference to the ods.metrics_credentials table.

metrics_credentials_detailed

Column Name	Description
id	The AUTO_INCREMENT generated primary key of the table. Note this value starts over with each run (sp_loadMetricsCredentialsDetailed) so you cannot count on this value being consistent between runs.
start	The unix timestamp for the start date of the record. **The time portion of the value is set to 00:00:00 US/Central timezone.**
end	The unix timestamp for the end date of the record. **The time portion of the value is set to 00:00:00 US/Central timezone.** When the record has no end date, the value is 0.

chunk_size	'day', 'week', or 'month'. The chunk_size determines the granularity of the time periods.
credential_id	The reference to the credentials table. This is the most important reference as it is the lowest level for which these metrics are recorded.
metrics_credentials_id	A reference back to the both the datamart.metrics_credentials and ods.metrics_credentials tables. Note the data will be consistent between this table and the datamart.metrics_credentials table, but it is expected that the ods.metrics_credentials table will continue to accumulate records and therefore have more data most of the time.
fetch_date	The fetch date for this record. This value comes from the metrics_credentials.date field. This field contains the actual value for the time as the application server calculated (it is not set to 00:00:00 US/Central).
record_type	This field is copied over from the metrics_credentials.record_type field.
first_record	This field is calculated by the sp_loadMetricsCredentialsDetailed procedure. The value is set to 1 if this is the first record for this credentials_id and type combination (ordered by fetch_date). The value is 0 otherwise.
service	The value for service is pulled from the credentials table.
type	The type of metric. The values are hardcoded in the php application. Types of metrics are unique given the service and the type (types are reused across services).
value	The value for the metric.
unique_id	The value for unique_id is pulled from the credentials table.
persona_id	The value for persona_id is pulled from the credentials table.
hash	md5(concat(`chunk_size`, ':', `service`, ':', `type`))

run_control

Column Name	Description
etl_name	Primary key.
start_time	The datetime on the database when the ETL last started. This field cannot be null.
status	The status of the ETL (Running or Complete). This field cannot be null.
end_time	The datetime on the database when the ETL last finished.