Virginia Tech Institute for Policy & Governance

VLDS Shaker Specifications

Aaron Schroeder
3/28/2013

Contents

1 Change History

Version #	Date	Entered by	Comments
1.0	12/5/2012	Aaron Schroeder	Initial Documentation
1.1	12/14/2012	Austin Mills	Added "FIELD_FREQ_IDS" table. Moved "MATCH_COLUMNS" table in the "Configuration" category.
2.0	3/1/2013	Austin Mills	Removed Demographic Log Reduction section and tables, added resumability section, updated processing table names, added updated shaker overview diagram
2.1	3/28/2013	Aaron Schroeder	Changed Hash Query to Identifiers Query. Updated web service descriptions with new design changes.

2 Terms

Exposure Database – Agency databases used in the VLDS that resides behind the agency firewall and contains secured demographic log records and detail tables

Lexicon – VLDS component that stores Metadata and Valid Values for exposed agency data. This data is stored outside the agency environment.

DataAdapter – Tool built on top of each agency's exposure database used to interpret queries and deliver secured records to the Shaker and back to researchers.

CRM – External tool that is used to submit data packages to the Shaker

Data Package – Collection of Data Requests (max of 7) used to accomplish a research purpose using agency data

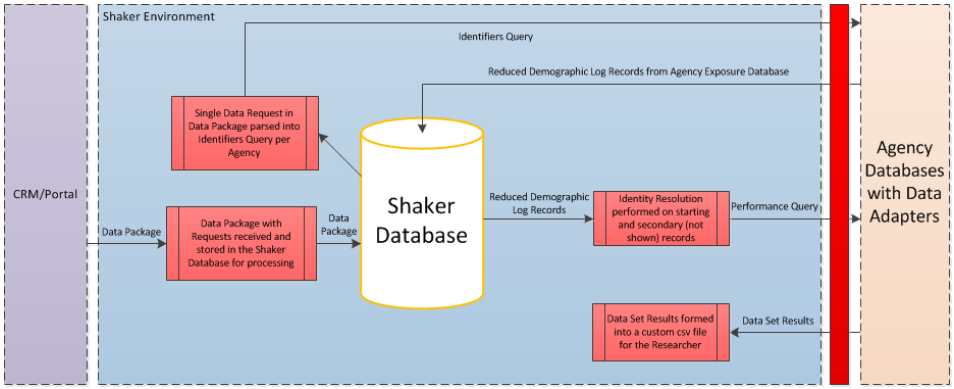
Data Request – Specific query of agency data from exposure databases. The request is parsed into an Identifiers and Performance Query (explained in this document).

3 Shaker Overview

The Shaker is one of the most critical and intricate components in the VLDS. At a high level, the Shaker is responsible for processing data packages and returning valuable data results to researchers in a secure manner. This document will explain how data packages with requests are processed and what is stored in the Shaker. Below is a diagram displaying the Shaker environment and how it interacts with other components in the VLDS with steps (Figure 1).

UPDATE DIAGRAM

Figure 1



~~The DataAdapter has a JDBC database connection to the Exposure DB.~~

~~Notes:~~

~~94.0 Table COLUMN_METADATA~~

~~14~~24 Processing a Data Package

This section will give details for each step in the Shaker for processing Data Packages and Requests. Each section will go in order from start (incoming Data Packages) to finish (final dataset results).

~~14.1~~4.1 Storing Data Packages

After a requested Data Package has been approved and sent from the CRM to the Shaker it is stored in the Shaker Database. All Data Packages are first stored in the Data Package Receipt table before validation so the Shaker keeps a record of all Packages. If the Data Package is invalid, a response would be sent back describing the errors and it would not be assigned a Receipt ID. All valid Data Packages are stored in the Data Package table in the Shaker. This table contains the XML of the data package with the name, description, and the location of the results once completed.

~~14.2~~4.2 Storing Data Requests

Each Data Package contains at least one Data Request and no more than 7. These requests are stored similarly as the packages in the Shaker database. The Data Request table houses the XML of the request along with the name, description, and SQL string.

~~14.3~~4.3 Executing Data Requests in a Package

Each Data Request inside a Data Package is processed one by one in the Shaker. After every Data Request (7 is the max) in the Package has been processed and the data result file has been created, the next Data Package in the queue can be sent through the Shaker. If a Data Request requires more than one associated agency data set, the starting and 1st associated agency is queried first, followed by the starting and 2nd associated agency. This is similar to how the data results are structured which can be seen here.

~~14.4~~4.4 Identifiers Query Execution

Each Data Request in a Package is parsed individually into an Identifiers query. The query (formerly known as Hash query) is submitted from the Shaker to the starting and secondary agency's Exposure Database through the DataAdapter. Only the Demographic Log records of individuals that apply to the Data Request are pulled. The records pulled from the exposure database are already hashed and reduced to eliminate any redundancies and improve processing. These records are then temporarily stored in the IDENTIFIERS 1 and IDENTIFIERS 2 tables of the Shaker.

Note: The Demographic Log Reduction process has recently been removed from the Shaker and now exists in the DataAdapter.

~~14.4.1~~4.4.1 Indexing Matching Tables

Indexing will be performed on the IDENTIFIERS tables in the Shaker Database. The following defines the indexes that will be applied to the two tables after they have been populated with the reduced Demographic Log records:

Figure 2

IDENTIFIERS_1	
PK	**SHAKER_TEMP_ID**
	PERSON_UNIQUE_ENTITY_ID
I1,I7,I5	PERSON_FIRST_NAME
	PERSON_MIDDLE_NAMES
I1,I8,I6	PERSON_LAST_NAME
I8,I7	PERSON_GENDER
	PERSON_DOB_DAY
	PERSON_DOB_MONTH
	PERSON_DOB_YEAR
I6,I5	PERSON_DOB_MONTH_YEAR
	PERSON_FIPS_5
	PERSON_PHONE
	PERSON_EMAIL
I2	PERSON_MATCH_ID_1
I3	PERSON_MATCH_ID_2
I4	PERSON_MATCH_ID_3
	CREATED_DATE

IDENTIFIERS_2	
PK	**SHAKER_TEMP_ID**
	PERSON_UNIQUE_ENTITY_ID
I7,I5,I1	PERSON_FIRST_NAME
	PERSON_MIDDLE_NAMES
I8,I6,I1	PERSON_LAST_NAME
I8,I7	PERSON_GENDER
	PERSON_DOB_DAY
	PERSON_DOB_MONTH
	PERSON_DOB_YEAR
I6,I5	PERSON_DOB_MONTH_YEAR
	PERSON_FIPS_5
	PERSON_PHONE
	PERSON_EMAIL
I2	PERSON_MATCH_ID_1
I3	PERSON_MATCH_ID_2
I4	PERSON_MATCH_ID_3
	CREATED_DATE

Note that the indexes are aligned with the blocking schemes used with probabilistic matching and also allow for improving the processing required for deterministic matching. The impacts of these indexes are evaluated using SQL Server query analysis tools to insure that they are having a positive impact on the matching performance.

~~4.2.2~~4.4.2 Index Classes

The following classes will be used to create the indexes after the data is populated:

IShakerDatabaseFacade – Contains methods specifically for creating the required indexes on the Identifier tables

Facade – Contains a method to call the database façade method to create the indexes.

DemographicLogQueryProcessor – Contains a call to the DataAdapter façade method to create the indexes.

~~4.5~~4.5 Identity Resolution

After the reduced Demographic Log records have been pulled for a Data Request, Identity Resolution takes place. Identity Resolution is the process where the Shaker performs matching on the Demographic Log identities from the starting and secondary agencies. This is a complex process that determines if a

unique entity is the same across the two agencies. There are two types of matching: Deterministic and Probabilistic. These methods are explained in the sections below.

~~4~~2.5.~~1~~4.5.1 Deterministic Matching

Deterministic Matching is the practice of pairing up records that match 100% across two agency Demographic Logs. This can easily be done if the records both have the same unique identifier (SSN, STI, etc.). For example, if John Doe in one agency had the same SSN as a John Doe in another agency, we can determine that they are indeed a match.

~~4~~2.5.~~2~~4.5.2 Probabilistic Matching

Probabilistic Matching is a process of comparing records from two data sets where it is uncertain if they are a complete 100% match. To figure out if the two records are a match, matching record fields (First Name, Last Name, DOB, and Gender to name a few) are compared. These fields are the ticket to determine if we can deduce that the two records indeed match. There are two things we need to know about these fields: how reliable is the data (m), and how common are the values (u). If a field is not reliable, it loses credibility to help with record matching. Also, if commonness of a field is too high, we cannot ultimately deduce that the records match by using that field.

For more information on Probabilistic Matching, please refer to the Probabilistic Matching Primer.

~~4~~2.6~~4~~4.6 Performance Query

After the Unique Identities have been matched, they are placed in the Shaker Probabilities table. This table lists out every match for the starting and secondary agencies and whether they were Deterministic or Probabilistic matches. The Shaker then creates a Performance Query of the Unique Entity IDs from the Probabilities table to send to the agencies and their exposure database detail tables.

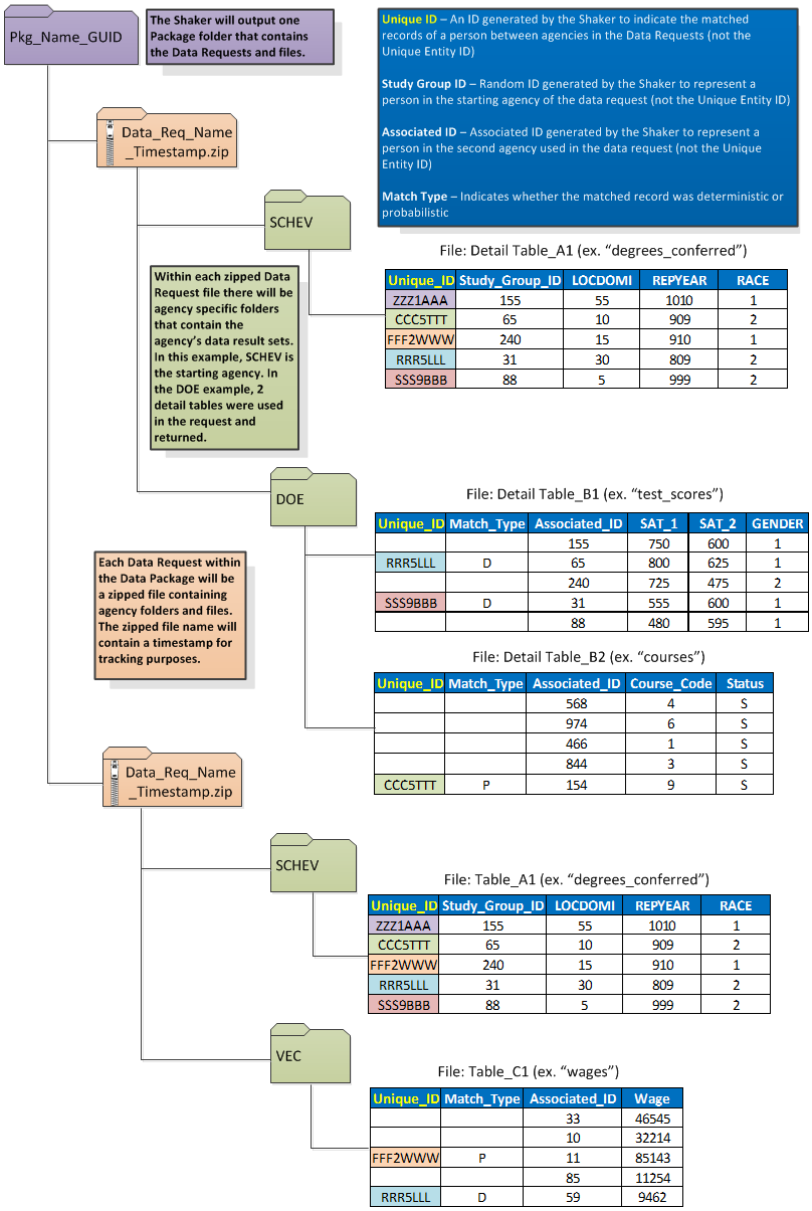
~~4~~2.7~~4~~4.7 Dataset Results

The Performance Query results are returned with the original specified fields in the Data Request and no Personal Identifiable Information (PII). When an entire Data Package is complete, all the Data Request results are placed in a Data Package folder. Inside that folder are the Data Requests as zip files. Those zip files contain the CSV results per agency. The CSV files will be separated if the size is too large. Figure 3 below better explains the dataset result structure from the Shaker.

Figure 3

DATA SET RESULTS STRUCTURE

This document represents the data result set output example from the Shaker after a data package has been processed and unmatched was selected.



Unique_ID	Study_Group_ID	LOCDOMI	REPYEAR	RACE
ZZZ1AAA	155	55	1010	1
CCC5TTT	65	10	909	2
FFF2WWW	240	15	910	1
RRR5LLL	31	30	809	2
SSS9BBB	88	5	999	2

Unique_ID	Match_Type	Associated_ID	SAT_1	SAT_2	GENDER
		155	750	600	1
RRR5LLL	D	65	800	625	1
		240	725	475	2
SSS9BBB	D	31	555	600	1
		88	480	595	1

Unique_ID	Match_Type	Associated_ID	Course_Code	Status
		568	4	S
		974	6	S
		466	1	S
		844	3	S
CCC5TTT	P	154	9	S

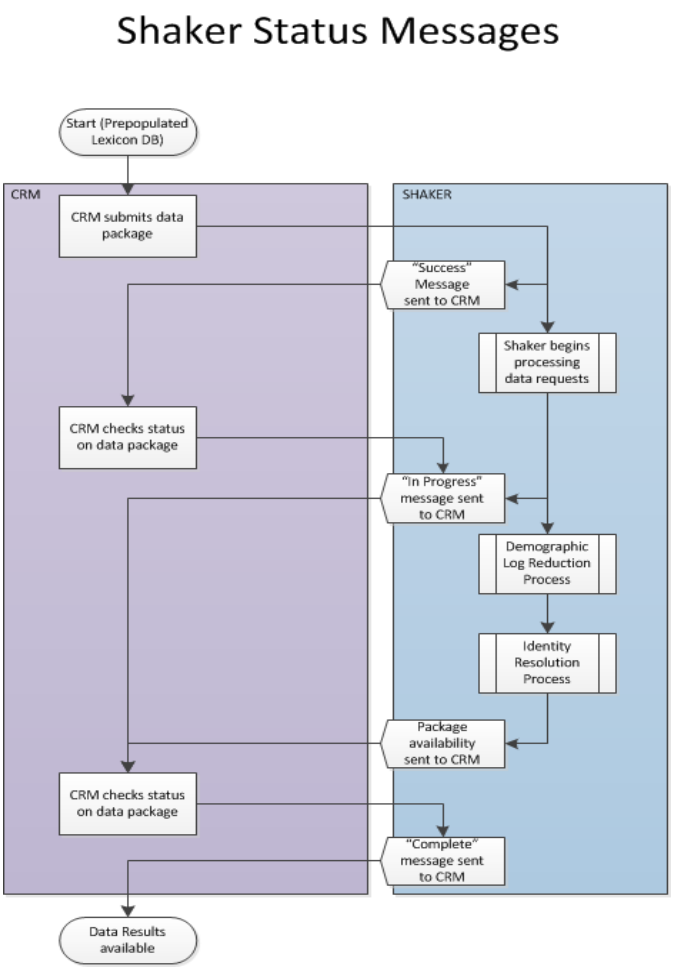
Unique_ID	Study_Group_ID	LOCDOMI	REPYEAR	RACE
ZZZ1AAA	155	55	1010	1
CCC5TTT	65	10	909	2
FFF2WWW	240	15	910	1
RRR5LLL	31	30	809	2
SSS9BBB	88	5	999	2

Unique_ID	Match_Type	Associated_ID	Wage
		33	46545
		10	32214
FFF2WWW	P	11	85143
		85	11254
RRR5LLL	D	59	9462

~~14.3.0 Metadata synchronization~~

Throughout the execution of Data Packages and Data Requests, the Shaker has the ability to send status messages back to the CRM. This is done using the ReportDataPackageStatus web service that is hosted by the Shaker. Depending on the size of the Data Package, the process can take hours or days to complete and get the final data result. In the meantime, the CRM can check on the status of the Data Packages to see if it has been started, the current progress, and if has finished or failed. The diagram below gives an example of the Shaker status messages during the execution of a Data Package:

Figure 4

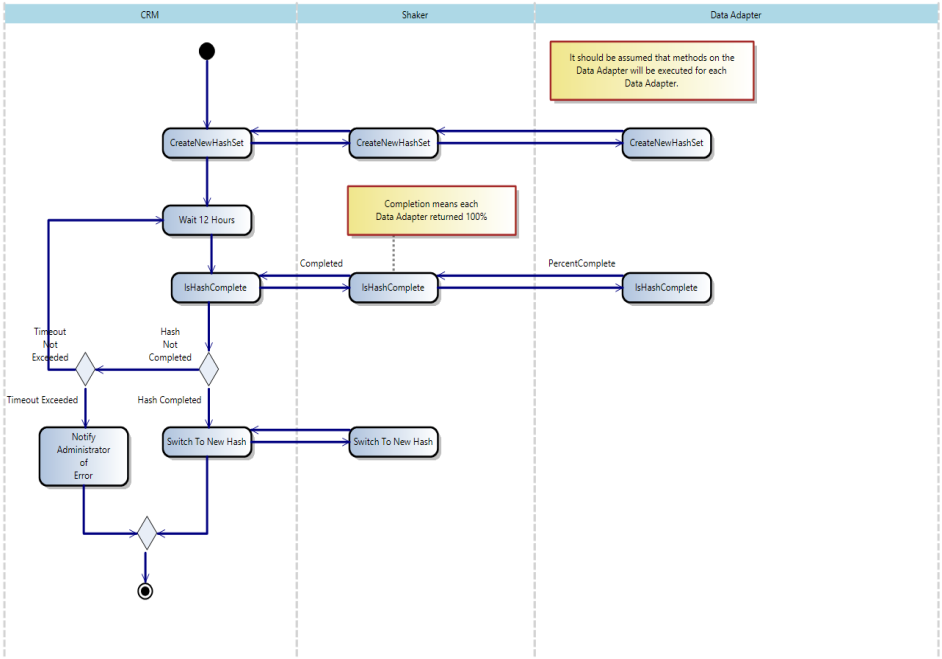


~~1486~~ CRM/Shaker Hashing Workflow

A CRM 2011 workflow will kick off the hash process every 90 days. This workflow will start the process and will poll the Shaker service every 12 hours to confirm that the hashing has completed. When the hashing has completed, it will call a method on the shaker to switch the DataAdapters to the new hashing algorithm. This method actually schedules the process rather than actually switching it.

This workflow will follow the process below (Figure 5). A custom workflow activity will be required to make the web service calls.

Figure 5



~~14~~97 Shaker Web Services

The Shaker hosts Windows web services used for interacting with the DataAdapter and CRM and consumes web services from the DataAdapter and Lexicon. These web services are critical for delivering a Data Package from the CRM and passing status messages back from the Shaker.

~~14.1~~7.1 Shaker as a Producer

The web services shown below are hosted by the Shaker and consist of interactions to the DataAdapter and CRM.

~~14.1.1~~7.1.1 Shaker Ping Web Service

Consumer: Most VLDS components

Description: This web service is simply used to test the connection to the Shaker to ensure it is up and running. The response should have HTTP status code of 200 (OK) and a plain text content type. The content should be empty.

~~14.1.2~~7.1.2 Shaker AcceptDataPackage Web Service

Consumer: CRM/Portal

Description: This web service is hosted by the Shaker allows the CRM to send Data Packages to the Shaker. **The Data Package contains the following:**

- *PackageId* – Guid uniquely identifying the Data Request package
- *Name* – Human readable identifier for the Data Request
- *DataRequests* – An array of Data Request objects representing the queries to be executed

The Data Requests within the Data Package contain the following:

- *DataRequestId* – Guid uniquely identifying this Data Request
- *IncludeUnmatched* – Specifies that for this data request, both matched and unmatched records should be returned
- *Name* – Human readable identifier for the data request
- *ReportID* – The Logi Report Id for the request
- *StartingAgency* – The agency whose master table is the start of the query
- *SelectedColumns* – An array of query field objects representing the columns that were selected as output for the data request
- *Filter* – A DataRequestFilterGroup object containing the filters
- *Query* – a string with the query from Logi to be used for troubleshooting purposes. This should never be shown to an agency sponsor or researcher since it will contain the internal column names.

~~149.1.37~~7.1.3 Shaker ReportDataPackageStatus Web Service

Consumer: CRM/Portal

Description: This web service allows the CRM to request a status update on the data package sent through the AcceptDataPackage web service. The update is sent back as a DataPackageResponse and DataRequestResponse.**The Data Package Response contains the following:**

- *PackageId* – Guid uniquely identifying the Data Request package
- *Completed* – Boolean indicating that the package processing is completed.
- *Responses* – An array of DataRequestResponse objects representing the responses for each corresponding DataRequest

The Data Request Response contains the following:

- *DataRequestId* – Guid uniquely identifying the Data Request being responded to.
- *DateStarted* – The date processing was started on the data request.
- *Status* – String with status or error message.
- *FileName* – Array of strings with the file name(s) containing the results.
- *HasError* – Indicates that an error occurred when processing the data request. If true, the status should contain the error message

~~149.1.47~~7.1.4 IShakerHashingService

Consumer: DataAdapter

Description: A web service interface will be exposed from the shaker to kick off the hash replacement and to validate that the hash process has completed.

System	Shaker
Service Name	CreateNewHashSet
Purpose	Triggers the process to create a new hash set.
Message specification	None

Message specification **Response**	Void
Notes	A new hash key will be generated and stored in the database using GetHMACKey.GetWheatHMACKey(positions, 3) for the hash and OneTimePadCipher.GetSecureRandomKey("ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789") for the anum salt. The dataadapters for each agency will be called to execute their CreateNewHashSet.

System	Shaker
Service Name	IsHashCompleted
Purpose	Queries to determine if the hash process has completed or is still in progress. The method returns a Boolean indicating it's completion status.
Message specification	None
Message specification **Response**	Complete, bool – true indicates that all dataadapters responded with 100% completion, false indicates that at least one agency is not finished
Notes	This calls the method in the dataadapter for each agency

System	Shaker
Service Name	ScheduleSwitchToNewHash
Purpose	Writes to the workflow tracking table that the system should switch to the new Hash. See Restart specification
Message specification	None
Message specification **Response**	None
Notes	The Dequeue method of the shaker workflow will will check for the need to switch hashes before kicking off package

System	Shaker
Service Name	SwitchToNewHash
Purpose	Sends the message to the dataadapters to switch to the new hash.
Message specification	None
Message specification **Response**	None
Notes	This method is actually called from the Dequeue method rather than externally. In the event of failure, it sends a notification through CRM. The method should also clean up any remaining tables in the Shaker.

~~149.2~~7.2 Shaker as a Consumer

The web services in this section are not provided by the Shaker. They are consumed (used by) the Shaker in some way from the CRM, Lexicon or DataAdapter.

~~149.2.1~~7.2.1 IdentifiersQuery Web Service

Producer: DataAdapter

Description: This service is used in the Shaker's matching process for querying hashed identifying information. The query comes in to the DataAdapter and retrieves the specific Unique Entity IDs in the Demographic Log with the associated records. The Unique Entity IDs and hashed fields are then sent to the Shaker where the Identity Resolution process takes place.

~~149.2.2~~7.2.2 PerformanceQuery Web Service

Producer: DataAdapter

Description: The Performance Query web service is used to gather all the Unique Entity IDs from the Shaker that was joined during the ID Mapper process. The DataAdapter then queries the exposed data sets for those IDs with by comparing the UEIDs with the Internal IDs in the reduced Demographic Log. The reduced Demographic Log table acts as a join table for querying the detail records. All selected fields associated with those Internal IDs from the detail tables are then returned to the Shaker.

~~149.2.3~~7.2.3 DA_Generate_UEID Web Service

Producer: DataAdapter

Description: This web service is used to generate random Unique Entity IDs for records in the Demographic Log. These Unique Entity IDs map to Internal IDs in agency Demographic Logs. These IDs must be randomly generated every so often for an added layer of security. This eliminates the ability to easily track records in the VLDS. The Shaker is responsible for calling this web service.

~~149.2.4~~7.2.4 GetLatestCachedSchema Web Service

Producer: Lexicon

Description: The GetLatestCachedSchema web service is hosted at the Lexicon and used to pull the latest database schema from the Lexicon. The Shaker calls this web service to validate that incoming Data Request query schemas match the Lexicon schema.

~~149.2.5~~7.2.5 NotifyQueryResponse Web Service

Producer: CRM/Portal

Description:This web service allows the Shaker to send Data Package responses to the CRM after the initial Data Package query has gone through the Shaker. The web service compliments the SH_ReportDataPackageStatus-WS by retrieving a response to the status update. The response is sent back as a DataPackageResponse and DataRequestResponse. **The Data Packapge Response contains the following:**

- *PackageId* – Guid uniquely identifying the Data Request package
- *Completed* – Boolean indicating that the package processing is completed.
- *Responses* – An array of DataRequestResponse objects representing the responses for each corresponding DataRequest

The Data Request Response contains the following:

- *DataRequestId* – Guid uniquely identifying the Data Request being responded to.
- *DateStarted* – The date processing was started on the data request.
- *Status* – String with status or error message.
- *FileName* – Array of strings with the file name(s) containing the results.
- *HasError* – Indicates that an error occurred when processing the data request. If true, the status should contain the error message

~~149.2.6~~7.2.6 IHashingService

Produced By: DataAdapter

Consumed By: Shaker

Description: A java servlet will be exposed from the DataAdapter. The IHashingService will expose the servlet functionality to the Shaker.

System	Dataadapter
Service Name	CreateNewHashSet
Purpose	Triggers the process to create a new hash set using the given key and anum salt.
Message specification	Calls the rebuildhash servlet with the following signature *Url format goes here once it is completed* RebuildDL.svc?command=rebuild&hash_salt=HashSalt&anum_salt=AnumSalt& **HashSalt** – string, The key to use to hash the data. **AnumSalt** – string, The key to use to execute the "One Time Pad" algorithm on the data **Algorithms –** The algorithms used to pad/chaff the demographic fields
Message specification **Response**	Void
Notes	See the FormatAlgorithms method of the DemographicLogQuery Generator for them implementation to format the algorithms parameter.

System	Dataadapter
Service	IsHashComplete

Name	
Purpose	Queries to determine if the hash process has completed or is still in progress. The method returns a completion percentage, 100 percent indicating that it has completed. This calls the DA_Status Servlet Da_status.svc
Message specification	None
Message specification **Response**	Complete, bool – indicates that the process is complete. If the status is IDLE, it is completed.
Notes	The status servlet returns XML similar to the following: <StatusResponse> <Task>xxx0</Task> <state>xxx1</state> ... </StatusResponse> The task element is used to determine completion. If it is set to IDLE, it is complete. An error response is returned in the event of an error.

System	Dataadapter
Service Name	SwitchToNewHash
Purpose	Notifies the dataadapters that they should switch to the new hash. This calls the Calls the rebuildhash servlet with the following signature RebuildDL.svc?command=switch
Message specification	None
Message specification **Response**	Acknowledged, bool – indicates that the dataadapter will use the new hash data from that point forward.
Notes	

~~15~~08 Resumability Design

The Shaker is currently being designed to allow a data package to resume from the last stopping point if an error occurs. This will be used to allow processing to continue after stopping and restarting the Shaker service. It will also be useful for troubleshooting issues by picking up exactly where processing left off, or to work around unexpected networking or environment issues which could occur with long running Data Packages.

The details of each package being processed will be stored in the database to allow for resumption after stopping the process. Also, the Shaker processing tables will be modified so that they are unique per data request as described here.

The resumption of a package is triggered by re-submitting the original package. A cancel data package method will be added to the Shaker to clean up the processing tables of a partially executed package. Alternatively, the cleanup method should be executed when the hashing changes.

Eventually, the portal will be modified to allow the researcher to request a Resume of a package or a Cancel Resume of a package.

~~15~~0.18.1 Implementation Summary

Resumablity will be implemented by creating a “steps” framework. The entire process will be broken into smaller steps using the command pattern. The commands will encapsulate all of the processing logic, replacing the “processor” classes and take over some responsibility from the facades.

When a data package is received, the processing steps will be built and saved to the database in the STEP table. Each step will have an input parameters class and a progress class relevant for that step. Steps will not load any configuration information from the web.config. Instead, the step building process will load all configuration parameters and pass them as input to each step. The input and progress classes for each step will be XML serializable so that they can be saved and loaded from the database. Steps can mark progress by writing to the Progress property. The steps framework will detect the change and write the progress to the database. If an error occurs in any step, progress is stopped, the error is recorded and a response is sent back to CRM. The error response communication will not be written as a step, and instead will be a part of the steps framework. If the Shaker subsequently accepts a data package with the same ID as a package that is resumable, processing will continue on that package.

If the Shaker service is stopped, processing for the current package will automatically resume with the last step marked as “In Progress”. If the step is resumable, it may have a progress object saved and would resume using the saved progress.

If no unhandled errors occur and the Shaker Service is not stopped, each step is executed until the final step is reached. The final step will notify CRM with the successful results.

Note: The re-try logic will continue to swallow errors up until the re-try threshold. This means that the package will only stop processing re-tryable code if the re-try threshold is reached which would cause

the step to throw an exception. The steps framework would catch this exception, allowing processing to resume and continue to re-try that step until it is successful or the re-try threshold is reached again.

~~15~~0.28.2 Database Resumability Processing

The Shaker database will house a STEP table to hold a sequence of steps for a package and store updates to the progress of each step as processing continues for that package.

Processing tables will also be modified to reflect the current data request running in the case that the process must be stopped. This allows other packages to continue processing while maintaining the state of a previous package for resumption later. The processing is stopped for the entire package as soon as any error is encountered.

Each one of the tables listed below will have the Request_ID of the data request appended to it:

Old Table Name	Current Table Name	Example Table Name
BLOCKING_MATCH	BLOCKING_MATCH_[RequestID]	BLOCKING_MATCH_ E23AEB5BB2284076B72FF4D2D2FAA3C6
BLOCKING_IDS	BLOCKING_IDS_[RequestID]	BLOCKING_IDS_ E23AEB5BB2284076B72FF4D2D2FAA3C6
PROBABILITIES	PROBABILITIES_[RequestID]	PROBABILITIES_ E23AEB5BB2284076B72FF4D2D2FAA3C6
IDENTIFIERS_1	IDENTIFIERS_[RequestID]_1	IDENTIFIERS_ E23AEB5BB2284076B72FF4D2D2FAA3C6_1
IDENTIFIERS_2	IDENTIFIERS_[RequestID]_2	IDENTIFIERS_ E23AEB5BB2284076B72FF4D2D2FAA3C6_2

~~15~~19 Shaker Tables

The VLDS Shaker contains multiple database tables used for configurations and processing. This section will list all tables with definitions and columns.

~~15~~1.19.1 Configuration Tables

All table structures listed in this section are configuration tables and used to hold valuable information and settings for the Shaker. These tables are not altered in any way during the processing of a data package.

~~15~~1.1.19.1.1 AGENCY_DEMOGRAPHIC_LOG_CONFIG

This table consists of information regarding each Agency's Demographic Log that resides in the Exposure Database. This information is used during the Hash (Demographic Log) query.

Column name	Data type	Description
AGENCY_DEMOGRAPHIC_LOG_CONFIG_ID	INT	Config ID for the table
AGENCY_NAME	NVARCHAR(50)	
DEMOGRAPHIC_LOG_SCHEMA_NAME	NVARCHAR(50)	Agency Demographic Log Schema Name

Column name	Data type	Description
DEMOGRAPHIC_LOG_TABLE_NAME	NVARCHAR(50)	Agency Demographic Log Table Name
DEMOGRAPHIC_LOG_TYPE	NVARCHAR(20)	Person or Place
DEMOGRAPHIC_LOG_FILTER_APPEND	NVARCHAR(2000)	Stores filters used or are mandatory before submitting data request
MAX_RESULTS	INT	Max results that can be returned for the Demographic Log to help with processing speed
CREATED_DATE	DATETIME	
MODIFIED_DATE	DATETIME	

~~15~~1.~~1.2~~9.1.2 BLOCKING_SCHEMES

This table holds information on the different blocking schemes used in the Shaker during Identity Resolution.

Column name	Data type	Description
BLOCKING_SCHEME_ID	INT	Table ID and Primary Key
BLOCKING_SCHEME_TYPE	NVARCHAR(20)	Type of blocking scheme
BLOCKING_SCHEME_ORDER	INT	
IDENTIFIER_1	NVARCHAR(80)	Starting agency identifier
IDENTIFIER_1_TRANSPOSE_FUNC	NVARCHAR(100)	
IDENTIFIER_2	NVARCHAR(80)	Secondary agency identifier
IDENTIFIER_2_TRANSPOSE_FUNC	NVARCHAR(100)	
CREATED_DATE	DATETIME	
MODIFIED_DATE	DATETIME	

~~151.1.3~~9.1.3 DATA_ADAPTER

This table is used to store information about each agency's DataAdapter that is installed in the agency environment. This information helps map the queries to the right agency DataAdapter during the processing of a data request.

Column name	Data type	Description
DATA_ADAPTER_ID	INT	Table ID and Primary Key
DATA_ADAPTER_NAME	NVARCHAR(50)	Name of the DataAdapter installed in the agency environment
DATA_ADAPTER_DESCRIPTION	NVARCHAR(2000)	Detailed description of the agency's DataAdapter
AGENCY_NAME	NVARCHAR(50)	Name of the agency where the DataAdapter is installed
SERVICE_URL	NVARCHAR(50)	DataAdapter URL
CREATED_DATE	DATETIME	
MODIFIED_DATE	DATETIME	

~~151.1.4~~9.1.4 DATA_PACKAGE_STATUS

This table holds Data Package status messages generated by the Shaker.

Column name	Data type	Description
PACKAGE_STATUS_CODE	NVARCHAR(50)	Generated status code
DESCRIPTION	NVARCHAR(50)	Data Package Description
CREATED_DATE	DATETIME	
MODIFIED_DATE	DATETIME	

~~151.1.5~~9.1.5 DATA_REQUEST_STATUS

This table holds Data Request status messages generated by the Shaker.

Column name	Data type	Description
REQUEST_STATUS_CODE	NVARCHAR(50)	Generated status code

Column name	Data type	Description
DESCRIPTION	NVARCHAR(2000)	Data Request Description
CREATED_DATE	DATETIME	
MODIFIED_DATE	DATETIME	

~~15~~1.1.6~~9~~9.1.6 DEMOGRAPHIC_LOG_FIELD_CONFIG

This table is used to store Demographic Field names as well as the hash algorithm used and comparison threshold.

Column name	Data type	Description
FIELD_NAME	NVARCHAR(50)	Demographic Log field name
DEMOGRAPHIC_LOG_TYPE	NVARCHAR(20)	Person or Place
HASH_ALGORITHM	NVARCHAR(50)	Name of the Hash Algorithm
COMPARISON_THRESHHOLD	FLOAT	Minimum number that must be met for two records to be a match
CREATED_DATE	DATETIME	
MODIFIED_DATE	DATETIME	

~~15~~1.1.7~~9~~9.1.7 DEMOGRAPHIC_LOG_HASH_ALGORITHM

The Hash Algorithm table consists of information for each Demographic Log hash used in the Shaker.

Column name	Data type	Description
DEMOGRAPHIC_LOG_HASH_ALGORITHM_NAME	NVARCHAR(50)	Name of the hash algorithm
DESCRIPTION	NVARCHAR(2000)	Hash algorithm description
CREATED_DATE	DATETIME	
MODIFIED_DATE	DATETIME	

~~151.1.8~~9.1.8 DEMOGRAPHIC_LOG_TYPE

This table lists whether the Demographic Log type is either a “Person” or “Place.” This table is not critical to the VLDS during the initial launch.

Column name	Data type	Description
DEMOGRAPHIC_LOG_TYPE_CODE	NVARCHAR(20)	Type code for a person or place
DESCRIPTION	NVARCHAR(2000)	Description of the Demographic Log type
CREATED_DATE	DATETIME	
MODIFIED_DATE	DATETIME	

~~151.1.9~~9.1.9 LEXICON_SCHEMA_XML

The Lexicon Schema XML table is used with the GetLatestCachedSchema web service hosted by the Lexicon. The Shaker stores numerous version of the Lexicon Schema in this table for comparisons when executing Data Requests.

Column name	Data type	Description
LEXICON_SCHEMA_XML_VERSION_ID	INT	Schema version ID
SCHEMA_XML	XML	Lexicon schema XML
CREATED_DATE	DATETIME	

~~151.1.10~~9.1.10 MATCH_DEMOGRAPHIC_LOG_CONFIG

This table is used during ID resolution for only probabilistic matching.

Column name	Data type	Description
MATCH_DEMOGRAPHIC_LOG_CONFIG_ID	INT	Configuration ID of the Demographic Log match
AGENCY_DEMOGRAPHIC_LOG_CONFIG_ID_1	INT	Starting agency configuration ID

Column name	Data type	Description
AGENCY_DEMOGRAPHIC_LOG_CONFIG_ID_2	INT	Secondary agency configuration ID
MATCH_THRESHOLD	FLOAT	Number the match must be above in order for it to be a probabilistic match
CREATED_DATE	DATETIME	
MODIFIED_DATE	DATETIME	

~~15~~1.1.~~11~~9.1.11 MATCH_TYPE

This table is used to describe the match types for Identity Resolution. The two types are Deterministic (D) and Probabilistic (P).

Column name	Data type	Description
MATCH_TYPE_CODE	NCHAR(1)	Match type (D = Deterministic, P = Probabilistic)
MATCH_TYPE_NAME	NVARCHAR(50)	Full name of Match type
CREATED_DATE	DATETIME	

~~15~~1.1.~~12~~9.1.12 PACKAGE_PROGRESS_INDICATOR

This table stores information about the progress of a Data Package.

Column name	Data type	Description
PACKAGE_PROGRESS_INDICATOR_CODE	NVARCHAR(50)	Indicator code
DESCRIPTION	NVARCHAR(2000)	Package progress description
CREATED_DATE	DATETIME	
MODIFIED_DATE	DATETIME	

~~15~~1.1.~~13~~9.1.13 PARAMETERS

The Parameters table is used to store Demographic Log matches of columns with their weights.

Column name	Data type	Description
PARAMETERS_ID	INT	Parameters ID
MATCH_DEMOGRAPHIC_LOG_CONFIG_ID	INT	Configuration ID of the Demographic Log match
COLUMN_M	FLOAT	M probability weight
COLUMN_U	FLOAT	U probability weight
COLUMNNAME	NVARCHAR(100)	Name of the Demographic Log column
CREATED_DATE	DATETIME	
MODIFIED_DATE	DATETIME	

~~15.1.14~~9.1.14 MATCH_COLUMNS

The Match Columns table lists out each Demographic Log column and how they can be matched (deterministically or probabilistically).

Column name	Data type	Description
MATCH_COLUMNS_ID	INT	Match Columns ID for the table
MATCH_COLUMN_1	NVARCHAR(50)	Demographic Log field
MATCH_COLUMN_2	NVARCHAR(50)	Demographic Log field
MATCH_TYPE_CODE	NCHAR(1)	Match type (D = Deterministic, P = Probabilistic)
MATCH_DEMOGRAPHIC_LOG_CONFIG_ID	INT	Configuration ID of the Demographic Log match
USE_IN_MATCH	NCHAR(1)	
CREATED_DATE	DATETIME	
MODIFIED_DATE	DATETIME	

~~15.1.15~~9.1.15 HASHKEY

A table will be added to store the hash key and anum salt being used. The hash key and anum salt will be stored in the database and a process will add a new key every 90 days. When adding the new key, it will

not initially be active (as indicated by the ACTIVE flag), and will not be made active until after it is confirmed that the DataAdapters have successfully hashed the data.

Note: The status of the DataAdapters hashing the data is not stored. The method to check the DataAdapter hash status will poll the DataAdapters at the time it is called, and will only consider it successful if all DataAdapters report 100%.

Column name	Data type	Description
HASH_ID	INT	Hash Key ID
HASH_KEY	NVARCHAR(50)	Hash Key used
ANUM_SALT	NVARCHAR(50)	Salt used
ACTIVE	BIT	Flag to indicate if the Hash Key is currently active
MODIFIED_DATE	DATETIME	

~~151.2~~9.2 Persistent Processing Tables

This section contains Shaker tables that are used primarily for processing of data packages and requests.

~~151.2.1~~9.2.1 DATA_PACKAGE_RECEIPT

When the Shaker first receives a Data Package, it is stored in this table. All Data Packages are stored here regardless of their validity. The entire Data Package is stored in the XML PACKAGE_SERIALIZED field.

Column name	Data type	Description
PACKAGE_RECEIPT_ID	NVARCHAR(50)	Receipt ID used by the CRM to track the Data Package
PACKAGE _ID	NVARCHAR(50)	Shaker Data Package ID – Invalid Packages will not have one
PACKAGE_NAME	NVARCHAR(100)	Data Package name
PACKAGE_SERIALIZED	XML	Entire Package unparsed
CREATED_DATE	DATETIME	

~~151.2.2~~9.2.2 DATA_PACKAGE

The DATA_PACKAGE table has information for valid Data Packages received by the Shaker.

Column name	Data type	Description
PACKAGE_ID	NVARCHAR(50)	Shaker Data Package ID
PACKAGE_RECEIPT_ID	NVARCHAR(50)	Receipt ID used by the CRM to track the Data Package
PACKAGE_NAME	NVARCHAR(100)	Data Package name
PACKAGE_DESCRIPTION	NVARCHAR(250)	Data Package description
PACKAGE_SERIALIZED	XML	Entire Package unparsed
PACKAGE_FOLDER_NAME	NVARCHAR(250)	Name of the folder that contains the Data Package results
PACKAGE_FOLDER_LOCATION	NVARCHAR(500)	Folder location that contains the Data Package results
PACKAGE_PROGRESS_INDICATOR_CODE	NVARCHAR(50)	
CREATED_DATE	DATETIME	
MODIFIED_DATE	DATETIME	

~~15~~1.2.~~3~~9.2.3 DATA_PACKAGE_RESPONSE

The Data Package Response table stores messages sent to the CRM about the status of a Data Package.

Column name	Data type	Description
DATA_PACKAGE_RECEIPT_ID	INT	Table ID and Primary Key
PACKAGE _ID	NVARCHAR(50)	
RESPONSE_SERIALIZED	XML	XML of the response back to the CRM
MESSAGE_TYPE	NVARCHAR(50)	Message type
CREATED_DATE	DATETIME	

~~1.2.4~~9.2.4 DATA_PACKAGE_STATUS_HISTORY

The status history table tracks what occurs to the Data Package as it goes through Shaker processing. Status messages may list out errors or when processing is complete.

Column name	Data type	Description
DATA_PACKAGE_ STATUS_HISTORY_ID	INT	Status History ID and primary key
PACKAGE _ID	NVARCHAR(50)	Shaker Data Package ID
STATUS_CODE	NVARCHAR(50)	
STATUS_MESSAGE	NVARCHAR(2000)	Detailed Data Package status message
CREATED_DATE	DATETIME	

~~1.2.5~~9.2.5 DATA_REQUEST_RECEIPT

When the Shaker first receives a Data Package, all of the Data Requests are stored in this table. The Data Request XML is stored in the REQUEST_SERIALIZED field.

Column name	Data type	Description
REQUEST_RECEIPT_ID	INT	Request Receipt ID
PACKAGE_RECEIPT _ID	NVARCHAR(50)	Package Receipt ID
REQUEST_ID	NVARCHAR(100)	Request ID assigned to each Data Request
REQUEST_NAME	NVARCHAR(100)	Request name
REQUEST_SERIALIZED	XML	Entire XML of the Request
CREATED_DATE	DATETIME	

~~1.2.6~~9.2.6 DATA_REQUEST

The DATA_REQUEST table has information for all Data Requests in Data Packages received by the Shaker.

Column name	Data type	Description
PACKAGE_ID	NVARCHAR(50)	Package ID that holds the Data Request

Column name	Data type	Description
REQUEST_ID	NVARCHAR(50)	Data Request ID and primary key
REQUEST_NAME	NVARCHAR(100)	Data Request name
REQUEST_DESCRIPTION	NVARCHAR(250)	Data Request description
REQUEST_SQL_STRING	TEXT	Data Request SQL string
REQUEST_SERIALIZED	XML	Entire Data Request unparsed
REQUEST_FILE_NAME	NVARCHAR(100)	Name of the file that contains the Data Request results
REQUEST_FILE_LOCATION	NVARCHAR(500)	File location that contains the Data Request results
CREATED_DATE	DATETIME	
MODIFIED_DATE	DATETIME	

~~151.2.7~~9.2.7 DATA_REQUEST_STATUS_HISTORY

The status history table tracks what occurs to the Data Request as it goes through Shaker processing. Status messages may list out errors or when processing is complete.

Column name	Data type	Description
DATA_REQUEST_ STATUS_HISTORY_ID	INT	Status History ID and primary key
REQUEST_ID	NVARCHAR(50)	Shaker Data Request ID
STATUS_CODE	NVARCHAR(50)	
STATUS_MESSAGE	NVARCHAR(2000)	Detailed Data Request status message
CREATED_DATE	DATETIME	

~~151.2.8~~9.2.8 DEMOGRAPHIC_LOG_QUERY

The Demographic Log (Hash) Query table in the Shaker contains information for the query such as the agency name, DataAdapter ID, and the SQL string.

Column name	Data type	Description
DEMOGRAPHIC_LOG_QUERY_ID	INT	Query ID and primary key
REQUEST_ID	NVARCHAR(50)	Data Request ID
DATA_ADAPTER_ID	INT	Agency DataAdapter ID
AGENCY_NAME	NVARCHAR(50)	Name of the participating agency
QUERY_TEXT	NVARCHAR(MAX)	Demographic Log query string
CREATED_DATE	DATETIME	

~~151.2.9~~9.2.9 DATASET_QUERY

The Dataset (Performance) Query table in the Shaker contains information for the query such as the agency name, DataAdapter ID, and the SQL string.

Column name	Data type	Description
DATASET_QUERY_ID	INT	Query ID
REQUEST_ID	NVARCHAR(50)	
DATA_ADAPTER_ID	INT	
AGENCY_NAME	NVARCHAR(50)	
QUERY_TEXT	NVARCHAR(MAX)	Data set query
CREATED_DATE	DATETIME	

~~151.3~~9.3 Volatile Processing Tables

This section contains Shaker tables that are used primarily for processing of data packages and requests. These tables are volatile because as data requests are being processed, data is temporarily stored until it is no longer needed and then deleted. This is a security feature as well so the Shaker doesn't permanently hold hashed PII data for very long.

~~151.3.1~~9.3.1 BLOCKING_IDS_[RequestID]

This is a temporary processing table for Probabilistic matching in ID resolution. The RequestID for the current Data Request will be appended at the end of the table name in case the request needs to be resumed.

Column name	Data type	Description
UNIQUE_ENTITY_ID1	NVARCHAR(100)	
UNIQUE_ENTITY_ID2	NVARCHAR(100)	
CREATED_DATE	DATETIME	

~~15~~1.3.2~~9~~9.3.2 _BLOCKING_MATCH_[RequestID]

This table is similar to the Identifier tables in the Shaker. The primary purpose of this table is to temporarily store matches performed during blocking. This takes place only during Identity Resolution. The RequestID for the current Data Request will be appended at the end of the table name in case the request needs to be resumed. Note: Place fields will not be used during the pilot phase.

Column name	Data type	Description
PERSON_UNIQUE_ENTITY_ID_1	NVARCHAR(50)	Table ID and Primary Key
PERSON_UNIQUE_ENTITY_ID_2	NVARCHAR(50)	
PERSON_FIRST_NAME	INT	
PERSON_LAST_NAME	INT	
PERSON_GENDER	INT	
PERSON_DOB_DAY	INT	
PERSON_DOB_MONTH	INT	
PERSON_DOB_YEAR	INT	
PERSON_DOB_MONTH_YEAR	INT	
PERSON_FIPS_5	INT	
PERSON_PHONE	INT	
PERSON_EMAIL	INT	
PERSON_MATCH_ID_1	INT	
PERSON_MATCH_ID_2	INT	

Column name	Data type	Description
PERSON_MATCH_ID_3	INT	
PLACE_UNIQUE_ENTITY_ID_1	INT	
PLACE_UNIQUE_ENTITY_ID_2	INT	
PLACE_NAME	INT	
PLACE_STREET_NUMBER	INT	
PLACE_STREET_NAME	INT	
PLACE_ZIP_CODE	INT	
PLACE_CONTACT_FIRST_NAME	INT	
PLACE_CONTACT_LAST_NAME	INT	
PLACE_PHONE	INT	
PLACE_EMAIL	INT	
PLACE_TYPE	INT	
PLACE_LATITUDE	INT	
PLACE_LONGITUDE	INT	
PLACE_MATCH_ID_1	INT	
PLACE_MATCH_ID_2	INT	
PLACE_MATCH_ID_3	INT	
CREATED_DATE	DATETIME	

~~15~~1.3.~~3~~9.3.3 DATA_PACKAGE_QUEUE

This table is populated when data packages are received by the Shaker and another data package is already being processed.

Column name	Data type	Description

Column name	Data type	Description
DATA_PACKAGE_QUE_ID	INT	Queue ID and primary key
PACKAGE_ID	NVARCHAR(50)	Data Package ID
CREATED_DATE	DATETIME	

~~1.3.4~~9.3.4 IDENTIFIERS_[RequestID]_1

The Identifiers 1 Table holds records for the starting agency's reduced Demographic Log before Identity Resolution. The RequestID for the current Data Request will be appended at the end of the table name in case the request needs to be resumed. Note: Place fields will not be used during the pilot phase.

Column name	Data type	Description
SHAKER_TEMP_ID	INT	Temporary Shaker ID for the record
PERSON_UNIQUE_ENTITY_ID	NVARCHAR(50)	Unique Entity ID randomly generated by the DataAdapter
PERSON_FIRST_NAME	NVARCHAR(80)	
PERSON_MIDDLE_NAMES	NVARCHAR(80)	
PERSON_LAST_NAME	NVARCHAR(80)	
PERSON_GENDER	NVARCHAR(50)	
PERSON_DOB_DAY	NVARCHAR(50)	
PERSON_DOB_MONTH	NVARCHAR(50)	
PERSON_DOB_YEAR	NVARCHAR(50)	
PERSON_DOB_MONTH_YEAR	NVARCHAR(50)	
PERSON_FIPS_5	NVARCHAR(200)	
PERSON_PHONE	NVARCHAR(50)	
PERSON_EMAIL	NVARCHAR(80)	
PERSON_MATCH_ID_1	INT	
PERSON_MATCH_ID_2	INT	

Column name	Data type	Description
PERSON_MATCH_ID_3	INT	
PLACE_UNIQUE_ENTITY_ID	NVARCHAR(50)	
PLACE_NAME	NVARCHAR(200)	
PLACE_STREET_NUMBER	NVARCHAR(50)	
PLACE_STREET_NAME	NVARCHAR(80)	
PLACE_ZIP_CODE	NVARCHAR(5)	
PLACE_CONTACT_FIRST_NAME	NVARCHAR(80)	
PLACE_CONTACT_LAST_NAME	NVARCHAR(80)	
PLACE_PHONE	NVARCHAR(50)	
PLACE_EMAIL	NVARCHAR(80)	
PLACE_TYPE	NVARCHAR(100)	
PLACE_LATITUDE	NUMERIC(10, 8)	
PLACE_LONGITUDE	NUMERIC(11, 8)	
PLACE_MATCH_ID_1	NVARCHAR(50)	
PLACE_MATCH_ID_2	NVARCHAR(50)	
PLACE_MATCH_ID_3	NVARCHAR(50)	
CREATED_DATE	DATETIME	

~~151.3.5~~9.3.5 IDENTIFIERS_[RequestID]_2

The Identifiers 2 Table holds records for the secondary agency's reduced Demographic Log before Identity Resolution with IDENTIFIERS_1. The RequestID for the current Data Request will be appended at the end of the table name in case the request needs to be resumed. Note: Place fields will not be used during the pilot phase.

Column name	Data type	Description

Column name	Data type	Description
SHAKER_TEMP_ID	INT	Temporary Shaker ID for the record
PERSON_UNIQUE_ENTITY_ID	NVARCHAR(50)	Unique Entity ID randomly generated by the DataAdapter
PERSON_FIRST_NAME	NVARCHAR(80)	
PERSON_MIDDLE_NAMES	NVARCHAR(80)	
PERSON_LAST_NAME	NVARCHAR(80)	
PERSON_GENDER	NVARCHAR(50)	
PERSON_DOB_DAY	NVARCHAR(50)	
PERSON_DOB_MONTH	NVARCHAR(50)	
PERSON_DOB_YEAR	NVARCHAR(50)	
PERSON_DOB_MONTH_YEAR	NVARCHAR(50)	
PERSON_FIPS_5	NVARCHAR(200)	
PERSON_PHONE	NVARCHAR(50)	
PERSON_EMAIL	NVARCHAR(80)	
PERSON_MATCH_ID_1	INT	
PERSON_MATCH_ID_2	INT	
PERSON_MATCH_ID_3	INT	
PLACE_UNIQUE_ENTITY_ID	NVARCHAR(50)	
PLACE_NAME	NVARCHAR(200)	
PLACE_STREET_NUMBER	NVARCHAR(50)	
PLACE_STREET_NAME	NVARCHAR(80)	
PLACE_ZIP_CODE	NVARCHAR(5)	
PLACE_CONTACT_FIRST_NAME	NVARCHAR(80)	

Column name	Data type	Description
PLACE_CONTACT_LAST_NAME	NVARCHAR(80)	
PLACE_PHONE	NVARCHAR(50)	
PLACE_EMAIL	NVARCHAR(80)	
PLACE_TYPE	NVARCHAR(100)	
PLACE_LATITUDE	NUMERIC(10, 8)	
PLACE_LONGITUDE	NUMERIC(11, 8)	
PLACE_MATCH_ID_1	NVARCHAR(50)	
PLACE_MATCH_ID_2	NVARCHAR(50)	
PLACE_MATCH_ID_3	NVARCHAR(50)	
CREATED_DATE	DATETIME	

~~151.3.6~~9.3.6 PROBABILITIES_[RequestID]

The PROBABILITIES table is used to store the matching records after Identity Resolution. It also records how the records were matched and the percentage probabilities. The RequestID for the current Data Request will be appended at the end of the table name in case the request needs to be resumed.

Column name	Data type	Description
UNIQUE_ENTITY_ID_1	NVARCHAR(100)	ID of the starting agency record
UNIQUE_ENTITY_ID_2	NVARCHAR(100)	ID of the secondary agency record
MATCH_TYPE_CODE	NCHAR(1)	Match type (D = Deterministic, P = Probabilistic)
BLOCKING_SCHEME_ID	INT	
PROB	FLOAT	Percentage probability of the Unique Entities matching
CREATED_DATE	DATETIME	

~~151.3.7~~9.3.7 STEP

The STEP table will be added to the database to hold the sequence of steps for a package and store updates to the progress of each step as processing continues for that package.

Column name	Data type	Description
STEP_ID	INT	An identity key for the step
STEP_TYPE	NVARCHAR(200)	System identifier for the step type
STEP_DESCRIPTION	NVARCHAR(200)	Human readable description of step
PACKAGE_ID	NVARCHAR(50)	The GUID of the package that the step is processing
SEQUENCE_NUM	INT	Sequence of the step within a package
CREATED_DATE	DATETIME	
INPUT	NVARCHAR(MAX)	Serialized XML of the progress for the step
INPUT_DESCRIPTION	NVARCHAR(200)	Short description of the input
PROGRESS	NVARCHAR(MAX)	Serialized XML of the progress
PROGRESS_DESCRIPTION	NVARCHAR(200)	Short description of the progress
STATUS	NVARCHAR(50)	Processing status of the step
MODIFIED_DATE	DATETIME	Date the step was updated