Yugandhar Open MDM Hub

Data Model Guide

Yugandhar Open MDM Hub Release – V2.0.0
Date – 11/06/2018

Contents

About Yugandhar Open MDM Hub Project

Master Data Management came a long way in last decade or so. There are currently more than 20 MDM solutions catering to various specializations of MDM like Customer Data Integration (CDI), Product Information Management (PIM), vendor and supplier management etc. However most of these solutions come with licensing costs amounting to thousands of dollar. To offer a completely free solution which would be made available through Apache 2.0 license, A Project is started in 2017 under the name 'Yugandhar Open MDM Project' to build Open Source MDM solutions catering to CDI, PIM and Data Governance Capabilities. Yugandhar in Sanskrit means Ever Lasting and the strongest of its time. Our vision is to build the strongest, Open Source, Multi Domain, Cross Industry and completely free MDM Solution.

We are happy to announce that the first release of the Yugandhar MDM Hub catering to CDI solution is built with Open source technologies like Spring and Hibernate etc, inbuilt data Model, 400+ ready to use services and having incredible Out of the Box capabilities is currently being distributed. We aim to make the current CDI offering the strongest and Planning to bring Data Stewardship and PIM solutions in upcoming years.

About this document

This document covers the data model of Yugandhar Open MDM Hub.

Understanding Data Table structure

The data entity table has below attributes in common. All the other attributes are used to store the data related to the entity however below attributes are mandatory for Yugandhar Code generation framework to work.

Column Name	Primary key	Nullable?	Data Type	Description
ID_PK	YES	No	VARCHAR2 (50 Byte)	ID to uniquely identify an entity in the system
VERSION		No	NUMBER	VERSION attribute used for optimistic lock
CREATED_TS		No	TIMESTAMP(6)	Creation timestamp of the record
DELETED_TS		YES	TIMESTAMP(6)	Soft-delete timestamp of the record
UPDATED_TS		No	TIMESTAMP(6)	The timestamp when the record is last updated
UPDATED_BY_USER		No	VARCHAR2 (50 Byte)	The userid which updated this record last
UPDATED_BY_TXN_ID		YES	VARCHAR2 (100 Byte)	The reference id of the transaction which updated the record

Understanding Reference Table structure

The reference data entity tables are meant to store key-value pairs. This is also called termed as 'List of Values' (LOB). LOV tables have some soft rules to easily identify them from other tables as below

1. The Name of reference data entity must start with 'REF_' characters
2. The CONFIG_LANGUAGE_CODE_KEY attribute must be present for multilingual support
3. The entity must have KEY and VALUE column.
4. By default, Yugandhar Open MDM Hub Reference data entity generated transactions does the search and matching on KEY attribute. The same should be followed going forward. However if needed the code can be modified to do the searching based on VALUE column as well.

5. The CONFIG_LANGUAGE_CODE_KEY and KEY attribute forms the unique constraint to restrict creating any duplicate values in the reference list.
6. The records of the reference data entities get cached in Yugandhar Open MDM Hub using ehcache caching framework. This is done to improve the performance of real time transactions. You have the option to disable the caching either by removing the cache statement or changing the cache expiry time to few milliseconds. However it's not advisable to disable the caching unless there is a strong reason to disable the same.

The reference data entity tables have below attributes in common.

Column Name	Primary key	Nullable?	Data Type	Description
ID_PK	Yes	N	VARCHAR2 (50 Byte)	ID to uniquely identify an entity in the system
VERSION		N	NUMBER	VERSION attribute used for optimistic lock
CREATED_TS		N	TIMESTAMP(6)	Creation timestamp of the record
DELETED_TS		Y	TIMESTAMP(6)	Soft-delete timestamp of the record
UPDATED_TS		N	TIMESTAMP(6)	The timestamp when the record is last updated
UPDATED_BY_USER		N	VARCHAR2 (50 Byte)	The userid which updated this record last
UPDATED_BY_TXN_ID		Y	VARCHAR2 (100 Byte)	The reference id of the transaction which updated the record

				The language code which provide multilingual support. This language code is refered from application configuration table
CONFIG_LANGUAGE_CODE_KEY	Unique Key	N	VARCHAR2 (50 Byte)	CONFIG_LANGUAGE_CODE. Not to be confused with REF_LANGUAGE_CODE, which gets used to store the list of languages to be used in conjunction with data like preferred language of the person.
KEY	Unique Key	N	VARCHAR2 (50 Byte)	The key of the key-value pair from list of values
VALUE		N	VARCHAR2 (50 Byte)	The value of the key-value pair from list of values
DESCRIPTION		Y	VARCHAR2 (50 Byte)	Description if any related to key value pair

Understanding Configuration tables

The configuration tables starts with prefix 'CONFIG_<entityname>'. The structure of the entity is similar to the data entities mentioned above. Also most of the configuration entities are cached, yet there is no fixed guideline on the same. The configuration tables are exclusively used by Yugandhar Framework and must not be customized by the user.

Column Name	Primary	Nullable?	Data Type	Description

	key			
ID_PK	YES	No	VARCHAR2 (50 Byte)	ID to uniquely identify an entity in the system
VERSION		No	NUMBER	VERSION attribute used for optimistic lock
CREATED_TS		No	TIMESTAMP(6)	Creation timestamp of the record
DELETED_TS		YES	TIMESTAMP(6)	Soft-delete timestamp of the record
UPDATED_TS		No	TIMESTAMP(6)	The timestamp when the record is last updated
UPDATED_BY_USER		No	VARCHAR2 (50 Byte)	The userid which updated this record last
UPDATED_BY_TXN_ID		YES	VARCHAR2 (100 Byte)	The reference id of the transaction which updated the record

Understanding Audit Log table structure

The Audit Log tables store the history Insert, update and delete operations performed on the base database entity. Audit log tables have some soft rules to easily identify them from other tables as below

1. The name of the Audit Log table starts with 'AL_<Name of the base entity>' e.g. the Audit table of the LEGALENTITY table is AL_LEGALENTITY. The name can be trimmed or changed if needed.
2. The Audit log table must have below mentioned attributes along with all the attributes of the base entity.
3. The AUDITLOG_ACTION_CODE gets updated as per below logic
   a. If a new record is created in base entity (e.g. LEGALENTITY) then audit log trigger will create a new record in corresponding audit log table (e.g. AL_ LEGALENTITY) with AUDITLOG_ACTION_CODE as 'I'.
   b. If a new record is updated in base entity (e.g. LEGALENTITY) then audit log trigger will create a new record in corresponding audit log table (e.g. AL_ LEGALENTITY) with AUDITLOG_ACTION_CODE as 'U'.
   c. If a new record is created in base entity (e.g. LEGALENTITY) then audit log trigger will create a new record in corresponding audit log table (e.g. AL_ LEGALENTITY) with AUDITLOG_ACTION_CODE as 'D'.
4. So Audit log tables will have multiple rows for each operation performed on every record giving a full audit history.
5. Please note that currently the Audit Log tables does not have any purging policy provided as the purging requirements are highly dissimilar so your Audit log data size may continue to rise. It is advisable to disable the audit history (by disabling relevant database triggers) if audit information is not needed or define a policy to purge in MDM project design phase itself.

Column Name	Primary key	Nullable?	Data Type	Description
AUDITLOG_ID_PK	YES	N	VARCHAR2 (50 Byte)	ID to uniquely identify an entity in the system
AUDITLOG_CREATED_TS		N	TIMESTAMP(6)	The timestamp when audit history record is created
AUDITLOG_ACTION_CODE		N	CHAR (1 Byte)	The action code of the audit

				log record based on base table

Understanding Application Configuration entities

CONFIG_APP_PROPERTIES - The table stores the application properties, should not be changed unless needed

CONFIG_ERRORCODE_REGISTRY –

This table stores all the error code and error messages being referred in MDM application

CONFIG_INQUIRY_LEVELS –

The table is used to configure the inquiry level related to any transaction.

CONFIG_LANGUAGE_CODE

Configuration Language Codes supported by Application

CONFIG_TXN_REGISTRY

The table is reference list storing the list of all the services available for Yugandhar MDM

Understanding Data Entities

Legal entity

Uses – Legal entity Model used to store the legal entities basic information. A Legal entity can be of Person (Individual) or Corporation (Organization) type. Based on the type of the legal entity the ENTITY_OBJECT_TYPE_REFKEY should be populated. There can only be an entry in Person or Corporation type of a given legal entity. i.e. Person and Corporation must never have same legal entity Idpk.

Database entities -

- LEGALENTITY
- LE_VEHICLE_ASSOC
- LE_TO_LE_RELATIONSHIP
- LE_SYSTEM_KEYS_REGISTRY
- LE_PROPERTY_ASSOC
- LE_PREFERENCES
- LE_PHONE_ASSOC
- LE_PERSON
- LE_IDENTIFIER_KYC_REGISTRY
- LE_CORPORATION
- LE_ADDRESS_ASSOC
- LE_ACCOUNT_ASSOC

Legal entity - Person

Uses – LE Person tables are used to store the person data

Database entities -

- LE_PERSON
- PERSONNAMES
- REF_PERSON_TYPE
- REF_PERSONNAME_TYPE

Legal entity - Corporation

Uses – LE Corporation tables are used to store the Corporation data

Database entities -

- CORPORATIONNAMES
- LE_CORPORATION
- REF_CORPORATION_NAME_TYPE
- REF_CORPORATION_TYPE

Legal entity - Address

Uses – LE Address tables are used to store the Address data for Person and Corporation type of entities.

Database entities -

- ADDRESS
- LE_ADDRESS_ASSOC
- REF_ADDRESS_SUBTYPE
- REF_ADDRESS_TYPE

Legal entity - Phone

Uses – Le Phone Model is used to store the phone, mobile, fax etc numbers of the Legal entity

Database entities -

- LE_PHONE_ASSOC
- PHONE_STANDARDIZED
- REF_PHONE_SUBTYPE
- REF_PHONE_TYPE

Account

Uses – Account Model is used to store Accounts of the Legal entity

Database entities - Tables of the Account Model include below database entities

- ACCOUNT
- ACCOUNT_ADDRESS_ASSOC
- ACCOUNT_PHONE_ASSOC
- LE_ACCOUNT_ASSOC
- REF_ACCOUNT_MDM_STATUS
- REF_ACCOUNT_SOURCE_STATUS

Account Phone

Uses – Account phone model is used to store the phone, mobile, fax etc numbers related to account.

Database entities -

- ACCOUNT_PHONE_ASSOC
- PHONE_STANDARDIZED
- REF_PHONE_SUBTYPE
- REF_PHONE_TYPE

Account Address

Uses – Account Address model is used to store the Address related to Account.

Database Entities -

- ACCOUNT_ADDRESS_ASSOC
- ADDRESS
- REF_ADDRESS_SUBTYPE
- REF_ADDRESS_TYPE

Entity Groups

Uses – Entity group's model is used to store the grouping of the entities

Database entities -

- ENTITY_GROUP
- ENTITY_GROUP_ASSOC
- REF_GROUP_SUBTYPE
- REF_GROUP_TYPE

LE Property

Uses – This model is used to store the property details related to entities

Database entities

- LE_PROPERTY_ASSOC
- PROPERTY
- REF_PROPERTY_LE_RELTYPE

LE Vehicle

Uses – This model is used to store the vehicle information related to legal entities.

Database entities -

- LE_VEHICLE_ASSOC
- VEHICLE

Identifier

Uses – This model is used to store the Identification number of the legal entities

Database entities -

- LE_IDENTIFIER_KYC_REGISTRY
- REF_IDENTIFICATION_TYPE

MISCELLANEOUS_INFO table

Uses – Miscellanious information is stored in this table.

Database entities **-**
MISCELLANEOUS_INFO

List of Reference Data (LOV) entities

- REF_ACCOUNT_MDM_STATUS
- REF_ACCOUNT_SOURCE_STATUS
- REF_ADDRESS_SUBTYPE
- REF_ADDRESS_TYPE
- REF_AGREEMENT_TYPE
- REF_ASSOC_TYPE
- REF_BATCH_ACTION_STATUS
- REF_BATCH_PROPOSED_ACTION
- REF_BILLING_MODE_TYPE
- REF_BRANCH_CODE
- REF_CLASSIFICATION_CODE
- REF_CORPORATION_NAME_TYPE
- REF_CORPORATION_TYPE
- REF_COUNTRY_ISO
- REF_CURRENCY
- REF_DEACTIVATION_REASON
- REF_ENTITY_OBJECT_TYPE
- REF_GENDER
- REF_GROUP_SUBTYPE
- REF_GROUP_TYPE
- REF_HIGHEST_EDU_QUAL
- REF_IDENTIFICATION_TYPE
- REF_IMPORTANCE_TYPE
- REF_INACTIVATION_REASON
- REF_INDUSTRY_CODE
- REF_LANGUAGE_CODE
- REF_LE_RATING
- REF_LE_RELATIONSHIP_TYPE
- REF_LE_ROLETYPE

- REF_LOBTYPE
- REF_MATCH_ACTIONSTATUS
- REF_MATCH_PROPOSED_ACTION
- REF_MATCH_RESULT
- REF_MATCH_SCORE
- REF_MATCH_THRESHOLD
- REF_MERGE_REASON
- REF_PERSON_TYPE
- REF_PERSONNAME_TYPE
- REF_PHONE_SUBTYPE
- REF_PHONE_TYPE
- REF_PREFERENCE_TYPE
- REF_PREFIX_NAME
- REF_PROPERTY_LE_RELTYPE
- REF_RELATIONSHIP_STATUS
- REF_SOURCE_SYSTEM
- REF_STATE_PROVINCE
- REF_STATUS_IN_SOURCE
- REF_STATUS_TYPE
- REF_SUFFIX_NAME
- REF_TERMINATION_REASON

List of Audit History Logging entities

- AL_ACCOUNT
- AL_ACCOUNT_ADDRESS_ASSOC
- AL_ACCOUNT_PHONE_ASSOC
- AL_ADDRESS
- AL_AUTH_ROLES_REGISTRY
- AL_AUTH_USER_REGISTRY
- AL_AUTH_USER_ROLE_ASSOC
- AL_AUTH_USERROLE_ACCESSCONTROL
- AL_BATCH_ENTITY_TO_PROCESS
- AL_CONFIG_APP_PROPERTIES
- AL_CONFIG_ERRORCODE_REGISTRY
- AL_CONFIG_INQUIRY_LEVELS
- AL_CONFIG_LANGUAGE_CODE
- AL_CONFIG_TXN_REGISTRY
- AL_CORPORATIONNAMES
- AL_ENTITY_GROUP
- AL_ENTITY_GROUP_ASSOC
- AL_INACTIVE_LE_REGISTRY
- AL_LE_ACCOUNT_ASSOC
- AL_LE_ADDRESS_ASSOC
- AL_LE_CORPORATION
- AL_LE_IDENTIFIER_KYC_REGISTRY
- AL_LE_PERSON
- AL_LE_PHONE_ASSOC
- AL_LE_PREFERENCES
- AL_LE_PROPERTY_ASSOC
- AL_LE_SYSTEM_KEYS_REGISTRY
- AL_LE_TO_LE_RELATIONSHIP
- AL_LE_VEHICLE_ASSOC

- AL_LEGALENTITY
- AL_MATCH_CANDIDATE_LE_REGISTRY
- AL_MATCH_MERGED_LE_ASSOC
- AL_MISCELLANEOUS_INFO
- AL_PERSONNAMES
- AL_PHONE_STANDARDIZED
- AL_PROPERTY
- AL_REF_ACCOUNT_MDM_STATUS
- AL_REF_ACCOUNT_SOURCE_STATUS
- AL_REF_ADDRESS_SUBTYPE
- AL_REF_ADDRESS_TYPE
- AL_REF_AGREEMENT_TYPE
- AL_REF_ASSOC_TYPE
- AL_REF_BATCH_ACTION_STATUS
- AL_REF_BATCH_PROPOSED_ACTION
- AL_REF_BILLING_MODE_TYPE
- AL_REF_BRANCH_CODE
- AL_REF_CLASSIFICATION_CODE
- AL_REF_CORPORATION_NAME_TYP
- AL_REF_CORPORATION_NAME_TYPE
- AL_REF_CORPORATION_TYPE
- AL_REF_COUNTRY_ISO
- AL_REF_CURRENCY
- AL_REF_DEACTIVATION_REASON
- AL_REF_ENTITY_OBJECT_TYPE
- AL_REF_GENDER
- AL_REF_GROUP_SUBTYPE
- AL_REF_GROUP_TYPE
- AL_REF_HIGHEST_EDU_QUAL
- AL_REF_IDENTIFICATION_TYPE

- AL_REF_IMPORTANCE_TYPE
- AL_REF_INACTIVATION_REASON
- AL_REF_INDUSTRY_CODE
- AL_REF_LANGUAGE_CODE
- AL_REF_LE_RATING
- AL_REF_LE_RELATIONSHIP_TYPE
- AL_REF_LE_ROLETYPE
- AL_REF_LOBTYPE
- AL_REF_MATCH_ACTIONSTATUS
- AL_REF_MATCH_PROPOSED_ACTION
- AL_REF_MATCH_RESULT
- AL_REF_MATCH_SCORE
- AL_REF_MATCH_THRESHOLD
- AL_REF_MERGE_REASON
- AL_REF_PERSON_TYPE
- AL_REF_PERSONNAME_TYPE
- AL_REF_PHONE_SUBTYPE
- AL_REF_PHONE_TYPE
- AL_REF_PREFERENCE_TYPE
- AL_REF_PREFIX_NAME
- AL_REF_PROPERTY_LE_RELTYPE
- AL_REF_RELATIONSHIP_STATUS
- AL_REF_SOURCE_SYSTEM
- AL_REF_STATE_PROVINCE
- AL_REF_STATUS_IN_SOURCE
- AL_REF_STATUS_TYPE
- AL_REF_SUFFIX_NAME
- AL_REF_TERMINATION_REASON
- AL_VEHICLE