Eagle ERD Overview

Eagle Investment Systems LLC

For technical support, contact us using any of the following methods:

One Wells Avenue
Newton, MA 02459
Phone: 617.219.0100
Fax: 617.558.9202
Toll Free: 800.380.5669

Eagle also maintains the following Internet sites:

http://www.eagleinvsys.com	Contains product information, company profile, and press releases
https://clientportal.eagleinvsys.com	

If you have comments or suggestions about this manual or find any errors in it, please contact us via e-mail at _Docs@eagleinvsys.com

Eagle ERD Overview

Printed in the USA
November 16, 2006 11:47 PM Edition

Table of Contents

Introduction

The Eagle ERD is an extensive and inclusive document that displays all of the tables within Eagle's databases. This ERD Overview document provides a simple, more descriptive view of certain tables within the ERD, focusing on the most frequently-used tables within the Eagle Data Model and diagramming key relationships among those tables. This document, in conjunction with the ERD Overview diagram beginning on page 11, will help you to further enhance your knowledge of the Eagle Data Model by understanding the unique identifiers required for uploader design.

Overview

The first page of the ERD Overview diagram (page 12) displays the major databases that are considered the focal point of most PACE implementations. The tables are linked together by a primary key. The primary key of a relational table uniquely identifies each record in the table. It can either be a normal attribute that is guaranteed to be unique (such as CUSIP), or it can be generated by the database management system (such as SECURITY_ALIAS in the SECURITY table). Primary keys may consist of a single attribute or multiple attributes in combination.

On this first page (page 12), the unique identifiers for securities (SECURITY_ ALIAS) and entities (ENTITY_ID) are displayed on the diagram. These unique identifiers are key factors in how data is linked among the major tables in the Eagle Data Model.

Security Alias: Unique Identifier for a Security

The SECURITY_ALIAS is the foundation of security data in PACE. This Primary Key is stored on the SECURITY_MASTER table of the SECURITY database. The SECURITY_ALIAS is a sequentially-assigned integer, created by the PACE_MASTERDBO.NEXT_ INSTANCE table and assigned to a security when the security is inserted into PACE. The SECURITY_ALIAS links security data to other major tables in the Eagle Data Model; thus, the first page of the diagram displays how the SECURITY_ALIAS joins the SECURITY_MASTER table to other security tables and to major tables that reside on other databases.

Entity ID: Unique Identifier for an Entity

The ENTITY_ID is the unique identifier for an Entity. It is limited to eight characters, and is either assigned by PACE (created by the PACE_ MASTERDBO.NEXT_INSTANCE table), or created via an uploader or through the Entity Maintenance functionality in PACE. If your organization requires more than eight characters to uniquely identify an entity, another field can be used to store the custom unique identifier, and the NEXT_ INSTANCE table can be used to populate the ENTITY_ID field.

The ENTITY_ID links entity data to other major tables in the Eagle Data Model. The bottom of the first page of the diagram (page 12) displays how the ENTITY_ID joins data from the ENTITY table to Cash, Trade, Holding, and Performance data.

Data Integrity

SECURITY_ALIAS and ENTITY_ID are also integral to maintaining the integrity of data that is imported into PACE. If you use the STAR to PACE functionality to load data into PACE, then data integrity is not compromised through the loading phase; however, if you bring data into PACE via uploaders, then the data will need to resolve to either an existing SECURITY_ALIAS or an existing ENTITY_ID. PACE will be unable to correctly retrieve the record in reports or exporters if it doesn't resolve to an existing SECURITY_ALIAS in the SECURITY_MASTER table, or an existing ENTITY_ID on the ENTITY table.

SECURITY Database

There are three different types of tables in the SECURITY database:

- Composite tables
- Historical tables
- Time series tables

Composite Tables and Historical Tables

The Composite tables store the most current composite view of security data, in which "composite" infers that each security record may be comprised of data from several different sources. The hierarchy of these sources is determined by the client loading the data. These master tables store only one record per security, which is referred to as the Best of Breed record. However, the associated Historical tables store security data differentiated by the source (SRC_INTFC_INST) and effective date (EFFECTIVE_DATE), and may have many records for a single security. This structure allows for rapid and efficient reporting if the report is designed to retrieve security composite data.

Best of Breed

The Best of Breed record is built using the Composite Security Engine. The record is based upon Composite Security Rules that you specify, often incorporating data from a variety of sources. This record is stored on the Composite tables (e.g., SECURITY_MASTER), as well as on the associated Historical tables (e.g., SECMASTER_HIST). The Historical tables include the original source security data as well as the Best of Breed record (which appears with the source of Eagle PACE).

Composite and Historical Tables Functionality

The SECURITY_MASTER table is the key table for the SECURITY database. There are tables on the SECURITY database that display attributes specific to the security's investment type (examples of such tables are EQUITY and FIXED_INCOME). Because these are Composite tables, they also have associated Historical tables, similar to the relationship between SECURITY_MASTER and SECMASTER_HISTORY. There are also a number of extension tables on the SECURITY database that store additional security reference data; SECURITY_MASTER_DETAIL and its associated Historical table, SECURITY_MASTER_DETAIL_HIST, are examples of such tables. Although overflow tables exist for the investment-type specific tables (e.g., EQUITY_DETAIL, MUNI_DETAIL), these tables are rarely populated.

The XREFERENCE table resolves the SECURITY_ ALIAS when there are multiple identifiers for a security (e.g., ISIN, SEDOL). This table is an elegant approach to resolving many types of security identifier issues. The alternative method uses a myriad of fields on the SECURITY_MASTER table, and is difficult

to maintain and to understand. The XREFERENCE_HIST table is the Historical table for the XREFERENCE table.

Time Series Tables

The PRICE, FX_RATES, ANALYTICS, and RATINGS tables are Time Series tables, in which the one table acts as both the main table and the Historical table for the data. These tables, as with all of the tables on the SECURITY database, are joined using the SECURITY_ALIAS field.

RULES Database

Entity ID

The unique identifier for an entity, the ENTITY_ID,links all entity data. The ENTITY_ID is the key to the entity structure because it links entity data among ENTITY tables, and also links such data with CASH, HOLDING, and TRADES tables.

Entity Tables

The ENTITY table on the RULES database stores information specific to each entity stored on the system, regardless of the type of entity. The ENTITY_TYPE column stores the category to which the entity belongs, whether it is a Portfolio, Composite, Aggregate, Sub-Portfolio, etc. The ENTITY_ EXTENSION table is used to store overflow data for entities. There are a number of fields on both the ENTITY and ENTITY_EXTENSION tables that may be customized to fit specific client needs. The ENTITY_ CHARACTERISTICS table allows you to store entity attributes by referencing a code and code values. The ENTITY, ENTITY_EXTENSION, and ENTITY_CHARACTERISTICS tables are similar to the SECURITY tables in that they all have related historical tables: ENTITY_HIST, ENTITY_ EXTENSION_HIST, and ENTITY_CHARACTERIS-TICS_HIST, respectively.

Entity Detail Tables

The ENTITY_DETAIL table stores details about entity relationships, including the structure of derived entities and an entity's associated benchmarks. Storing the definitions of derived entities provides a record of the entities' constituents for the entity build process, which requires this information to build holdings data for composites, aggregates, and sub-portfolios.

The ENTITY_DETAIL_HIST table is the associated historical table, which allows PACE to report off of these entities historically. For list entities, the ENTITY_LIST table utilizes codes and code values from the Data Steward to display the list's child entities. The code displays the types of entities held by the list, and the code value displays the ENTITY_ID of the child entity.

Entity Cross Reference Table

Similar to using the XREFERENCE table on the SECURITY database, you are encouraged to use the ENTITY_XREFERENCE table to allow storage of multiple identifiers for an entity. If you have multiple IDs for an entity, it is more straightforward to have one central location to resolve the ENTITY_ID.

Client Tables

Client tables store Client Relationship data that is often used for the Client Reporting module of PACE. The structure of these tables is similar to that of

the entities tables, in that there is a key table, CLIENT, and an associated Historical table, CLIENT_HIST. The CLIENT table stores attributes specific to clients, including the INSTANCE, the unique identifier for a client. Clients can be managers, institutions, account owners, etc. The CLIENT_RELATIONSHIPS table stores information about the type of relationships the client has, whether it is a client-to-client relationship or an entity-to-client relationship. PACE supports one-to-many relationships, so a client can have relationships with multiple entities or multiple clients (e.g., a client could be the accountant for many entities). Lastly, the CLIENT_ADDRESS table stores information about the client's address; you can store multiple addresses per client (e.g., Home, Business).

HOLDING Database

The HOLDING Database stores position data that is loaded into PACE.

Position Table

The POSITION table is the most important table in the HOLDING database; it stores information at the net portfolio level (e.g., Fund Market Value, Total Par Value). The effective date is recorded on this table; thus, an unlimited number of historical positions can be stored. The POSITION table can store multiple sources for any given effective date for an entity. The primary key on the POSITION table is POSITION_ID, which is created when the record is inserted into the table.

Position Detail Tables

Positions are stored at the security level on the POSITION_DETAIL table. The primary key on the POSITION_DETAIL table is POSITION_DETAIL_ID, which is created when a record is inserted into the POSITION_DETAIL table. There are a number of overflow tables at the net security level on the HOLDING database, including the POSITION_COST_DETAIL, POSITION_ INCOME_ DETAIL, and POS_COST_DETAIL_USER tables.

Lot Level Position Tables

The LOT_LEVEL_POSITION table stores security holding data at the lot level. This table represents holdings for each security, for a particular effective date, based on purchases and sales of the security. The primary key on the LOT_LEVEL_POSITION table is LOT_LEVEL_POSITION, which is created when a record is inserted into the table. The POSITION_INCOME_ LOT, POSITION_COST_LOT, and POSITION_COST_LOT_USER tables all serve to store overflow data at the lot level.

Position ID

The POSITION_ID is the unique identifier for a position; it links the POSITION table to the POSITION_DETAIL table and to the LOT_LEVEL_POSITION table. There is no link between the lot level and position detail level at the database level. The link between the various lot level holding tables is the LOT_ LEVEL_POSITION.

Loading Data into PACE

When loading holding data into PACE, the ENTITY_ID must resolve properly to the ENTITY table and the SECURITY_ALIAS must resolve properly to the SECURITY_MASTER table.

PERFORM Database

The PERFORM database stores all Performance data that is uploaded into PACE, or that is calculated by the PACE Performance module via Performance Calculation reports. The anchor table is the PERF_SUMMARY table, which stores summary level data. On the PERF_SUMMARY table, one summary record is stored per given effective date. The PERF_SUM_INST is stored on the PERF_SUMMARY table and is like the POSITION_ID on the HOLDING database; it is the unique instance for entity, date, performance model, type, level, and source.

The PERF_SEC_RETURNS table stores rollup and security level returns for each node of a performance model. There may be multiple performance returns stored for a given entity for a given effective date (e.g., Daily or Monthly Returns).

Dictionaries

Dictionaries play two important parts in Eagle PACE. They define a structured grouping for reports and performance calculations and exist as two distinct types:

- **Regular dictionaries** are user-defined categorization schemes, created in PACE, that use a tool for building groupings.
- **Performance dictionaries** are portfolio groupings set up for calculating performance returns.

When committing returns to the Performance database, each set of returns is uniquely identified by an instance. This instance is referred to as PERF_SEC_INST; it is found on the PERF_SUMMARY and PERF_SEC_RETURNS tables. Part of the key for this instance is the DICTIONARY_ID on the PERF_SUMMARY table.

Each performance return is linked to a Performance Model, which is identified by the DICTIONARY_ID. Dictionaries are stored on the RULES database, and those dictionaries for use in Eagle Performance have a Dictionary Type of "P." Each dictionary may have multiple levels of information, and each level is defined by the DICTIONARY_FORMAT on the PACE_MASTER database. The DICTIONARY_ FORMAT table stores the levels; each level is typically defined by a field attribute. For example, the format can be defined by three levels:

Entity ID

Country

Security Type

The DICTIONARY_DETAIL table further defines the model by nodes. A node is a subgroup of a level, and there can be nodes under any of the levels. The DICTIONARY_DETAIL table for this example has a row populated for each country present in the report. **Country** is a sub-level of **Entity ID**, and each country (in parentheses) is a node under the **Country** sub-level. **Security**

Type is sub-level of **Country**, and each security type (in parentheses) is a node under the **Security Type** sub-level:

Entity ID

Country

(USA)

(United Kingdom)

(Netherlands)

Security Type

(Common Stock)

(Municipal Bond)

CASH and TRADES Databases

Cash Database

The CASH database stores all cash activity, cash balances, and cash flow projection data. The CASH_ACTIVITY table stores the daily activity that affects the cash for an entity. The CASH_INT, stored on the CASH_ ACTIVITY table, is a unique combination of the SRC_INTFC_INST, EFFECTIVE_DATE, ENTITY_ID, SECURITY_ALIAS, and TRANS_NUM. Performance calculations rely on cash activity data and often call for this data.

The CASH_BAL table store daily cash balances for an entity. Typically balances are stored for each currency. The primary key for the CASH_BAL table is the CASH_BAL_INST, which represents a combination of SRC_INTFC_INST, EFFECTIVE_DATE, ENTITY_ID and SECURITY_ALIAS.

The CASH_FLOW_PROJECTION table is not frequently utilized. The CASH_FLOW_PROJECTION table was designed to store the positive and negative cash flows for an entity. The CASH_FLOW_INST is the primary key on the CASH_FLOW_PROJECTION table and represents what is required for a unique record, ENTITY_ID, PROJ_EFFECTIVE_DATE, EFFECTIVE_DATE, SRC_INTFC_INST, SECURITY_ALIAS and TRANS_NUM.

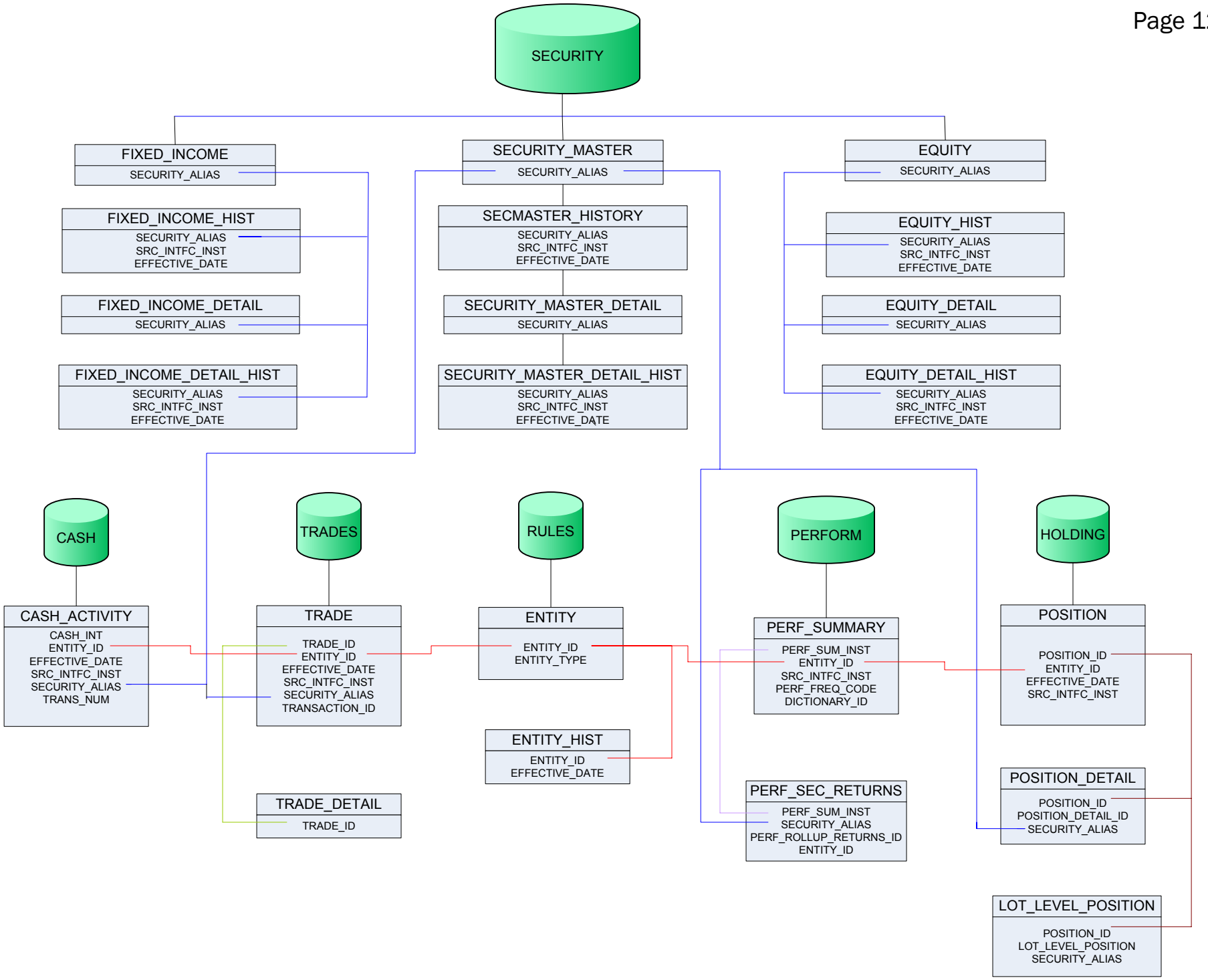
Trades Database

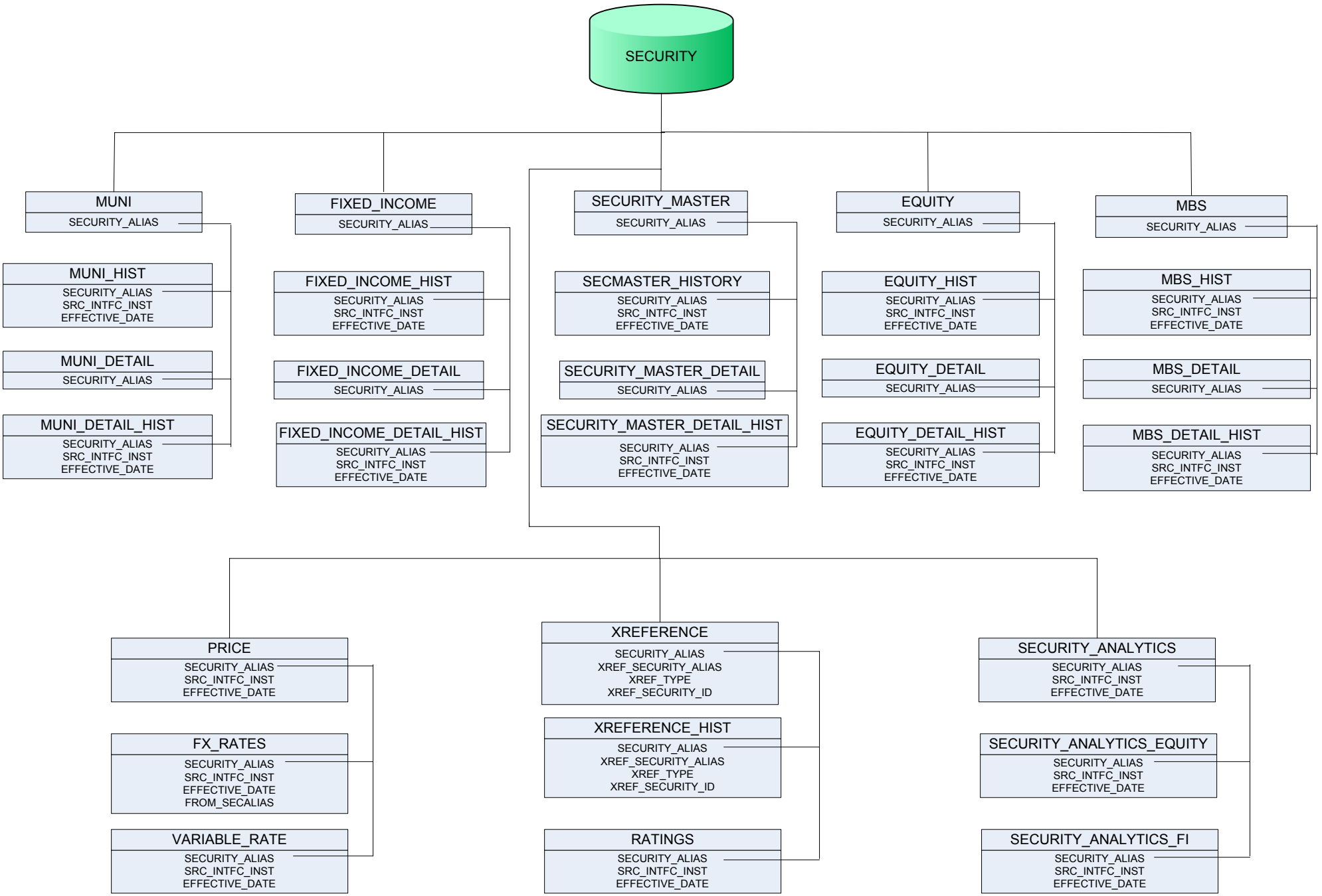
The TRADES database stores all trade transaction information. The main tables utilized on this database are TRADE and TRADE_DETAIL. The TRADE table is the anchor table for the TRADES database, while the TRADE_DETAIL table acts as an overflow table. Trade data is loaded into PACE either via an uploader, using Message Center, or by using STAR to PACE. STAR-generated transaction data is not stored in the TRADES database, but is stored in the eSTAR database. STAR will draw upon the CORP_ACT_HIST table for reference data regarding corporate action announcements.

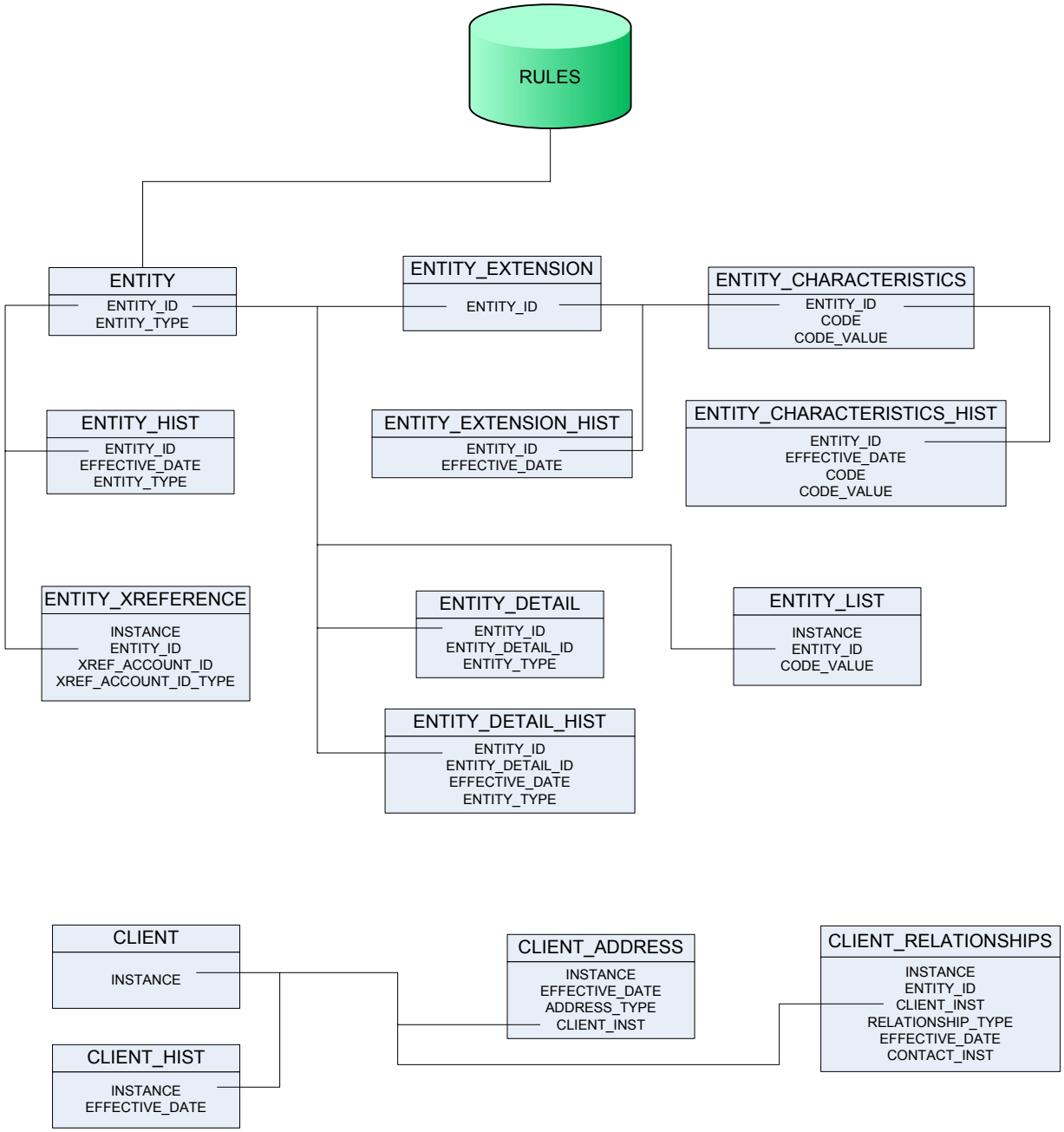
The TRADE_ID is a sequentially assigned integer, designated by PACE, that links the TRADE and TRADE_DETAIL tables. The TRADE_ID is based on the incoming record and represents a combination of ENTITY_ID, EFFECTIVE_DATE, SRC_INTFC_INST, SECURITY_ALIAS, and TRANSACTION_ID. The advantage of the TRADE_ID is that it is unique; PACE does not use transaction IDs from other systems because PACE cannot verify that they are unique, as it is possible to have the same Transaction ID generated by two different systems for the same trade.

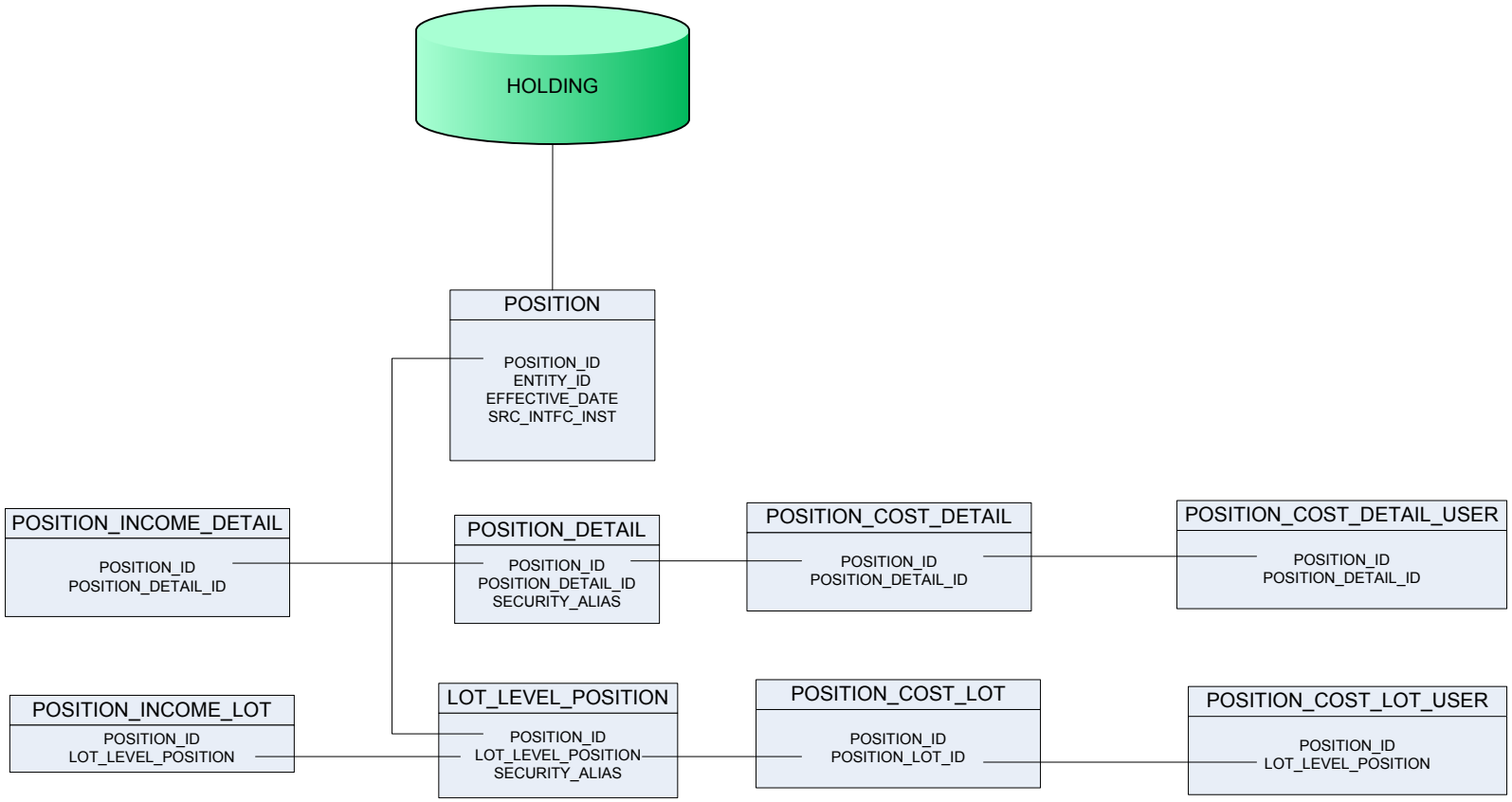
ERD Schematic Diagram

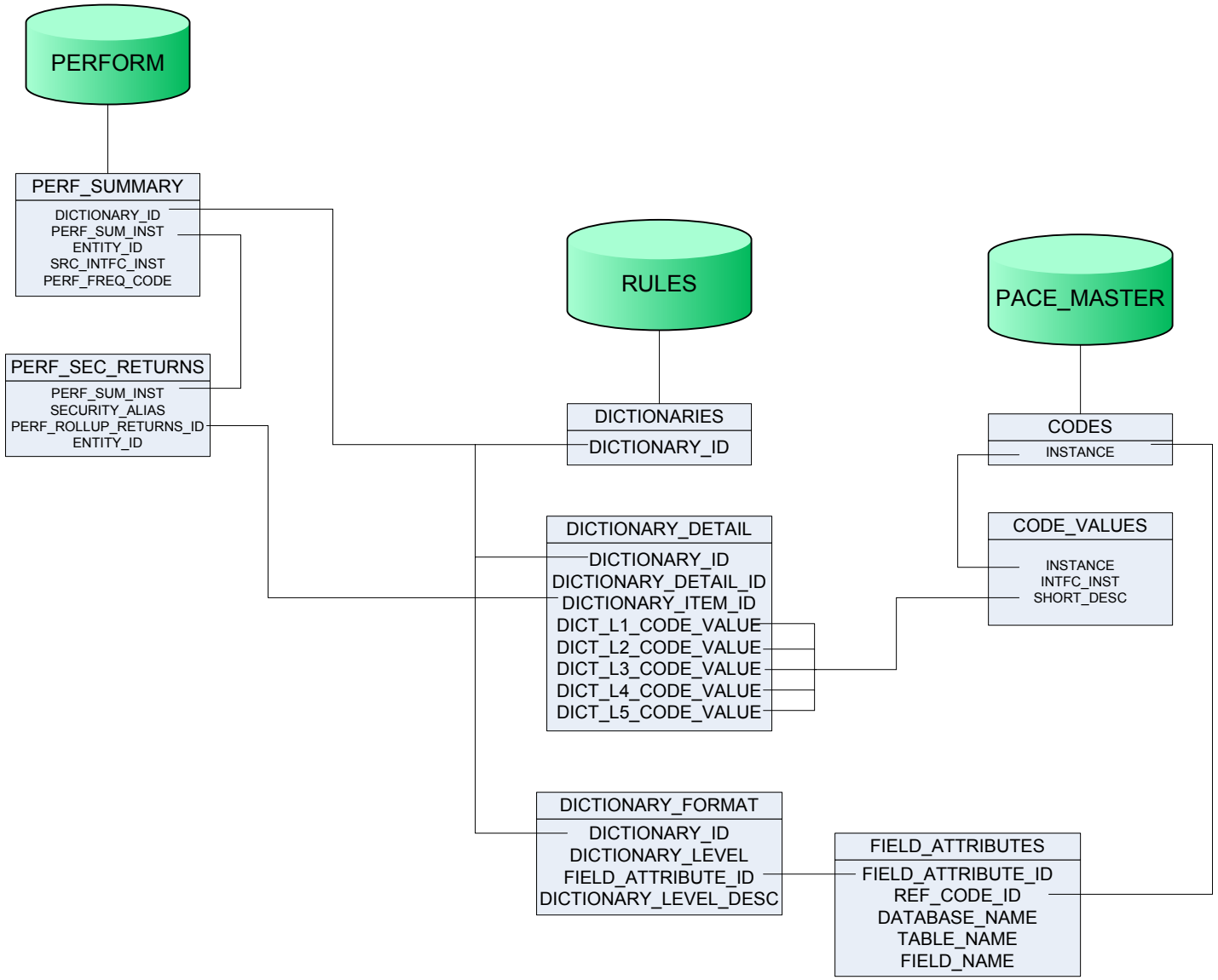
The following pages contain an abbreviated schematic diagram of the Eagle ERD. It illustrates the relationships between commonly-used tables to help you construct queries while designing or testing data uploads and exports.

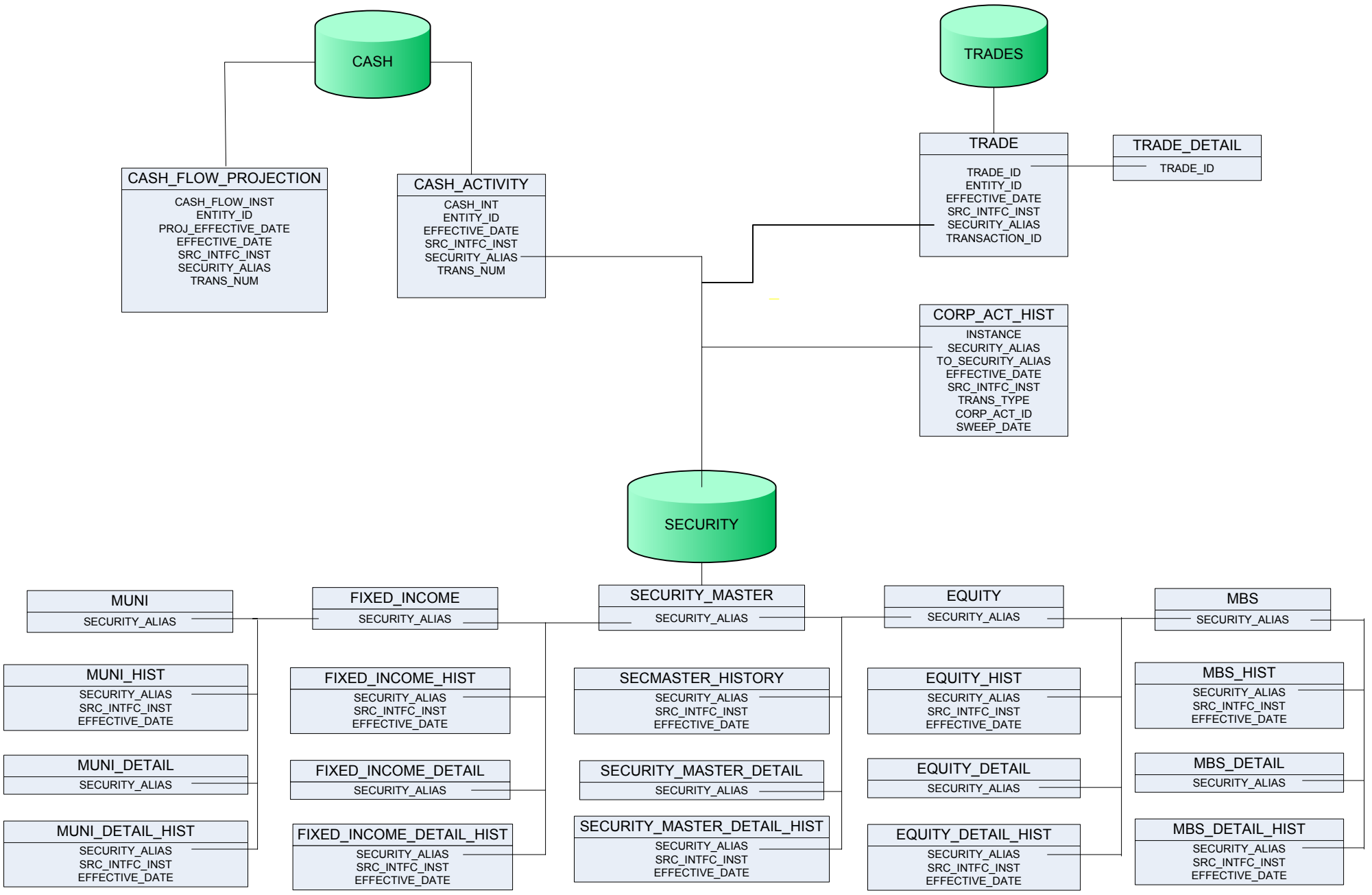

SECURITY
FIXED_INCOME
SECURITY_ALIAS
SECURITY_MASTER
SECURITY_ALIAS
EQUITY
SECURITY_ALIAS
FIXED_INCOME_HIST
SECURITY_ALIAS
SRC_INTFC_INST
EFFECTIVE_DATE
SECMASTER_HISTORY
SECURITY_ALIAS
SRC_INTFC_INST
EFFECTIVE_DATE
EQUITY_HIST
SECURITY_ALIAS
SRC_INTFC_INST
EFFECTIVE_DATE
FIXED_INCOME_DETAIL
SECURITY_ALIAS
SECURITY_MASTER_DETAIL
SECURITY_ALIAS
EQUITY_DETAIL
SECURITY_ALIAS
FIXED_INCOME_DETAIL_HIST
SECURITY_ALIAS
SRC_INTFC_INST
EFFECTIVE_DATE
SECURITY_MASTER_DETAIL_HIST
SECURITY_ALIAS
SRC_INTFC_INST
EFFECTIVE_DATE
EQUITY_DETAIL_HIST
SECURITY_ALIAS
SRC_INTFC_INST
EFFECTIVE_DATE
CASH
TRADES
RULES
PERFORM
HOLDING
CASH_ACTIVITY
CASH_INT
ENTITY_ID
EFFECTIVE_DATE
SRC_INTFC_INST
SECURITY_ALIAS
TRANS_NUM
TRADE
TRADE_ID
ENTITY_ID
EFFECTIVE_DATE
SRC_INTFC_INST
SECURITY_ALIAS
TRANSACTION_ID
ENTITY
ENTITY_ID
ENTITY_TYPE
PERF_SUMMARY
PERF_SUM_INST
ENTITY_ID
SRC_INTFC_INST
PERF_FREQ_CODE
DICTIONARY_ID
POSITION
POSITION_ID
ENTITY_ID
EFFECTIVE_DATE
SRC_INTFC_INST
ENTITY_HIST
ENTITY_ID
EFFECTIVE_DATE
TRADE_DETAIL
TRADE_ID
PERF_SEC_RETURNS
PERF_SUM_INST
SECURITY_ALIAS
PERF_ROLLUP_RETURNS_ID
ENTITY_ID
POSITION_DETAIL
POSITION_ID
POSITION_DETAIL_ID
SECURITY_ALIAS
LOT_LEVEL_POSITION
POSITION_ID
LOT_LEVEL_POSITION
SECURITY_ALIAS


SECURITY
MUNI
SECURITY_ALIAS
MUNI_HIST
SECURITY_ALIAS
SRC_INTFC_INST
EFFECTIVE_DATE
MUNI_DETAIL
SECURITY_ALIAS
MUNI_DETAIL_HIST
SECURITY_ALIAS
SRC_INTFC_INST
EFFECTIVE_DATE
FIXED_INCOME
SECURITY_ALIAS
FIXED_INCOME_HIST
SECURITY_ALIAS
SRC_INTFC_INST
EFFECTIVE_DATE
FIXED_INCOME_DETAIL
SECURITY_ALIAS
FIXED_INCOME_DETAIL_HIST
SECURITY_ALIAS
SRC_INTFC_INST
EFFECTIVE_DATE
SECURITY_MASTER
SECURITY_ALIAS
SECMASTER_HISTORY
SECURITY_ALIAS
SRC_INTFC_INST
EFFECTIVE_DATE
SECURITY_MASTER_DETAIL
SECURITY_ALIAS
SECURITY_MASTER_DETAIL_HIST
SECURITY_ALIAS
SRC_INTFC_INST
EFFECTIVE_DATE
EQUITY
SECURITY_ALIAS
EQUITY_HIST
SECURITY_ALIAS
SRC_INTFC_INST
EFFECTIVE_DATE
EQUITY_DETAIL
SECURITY_ALIAS
EQUITY_DETAIL_HIST
SECURITY_ALIAS
SRC_INTFC_INST
EFFECTIVE_DATE
MBS
SECURITY_ALIAS
MBS_HIST
SECURITY_ALIAS
SRC_INTFC_INST
EFFECTIVE_DATE
MBS_DETAIL
SECURITY_ALIAS
MBS_DETAIL_HIST
SECURITY_ALIAS
SRC_INTFC_INST
EFFECTIVE_DATE
PRICE
SECURITY_ALIAS
SRC_INTFC_INST
EFFECTIVE_DATE
FX_RATES
SECURITY_ALIAS
SRC_INTFC_INST
EFFECTIVE_DATE
FROM_SECALIAS
VARIABLE_RATE
SECURITY_ALIAS
SRC_INTFC_INST
EFFECTIVE_DATE
XREFERENCE
SECURITY_ALIAS
XREF_SECURITY_ALIAS
XREF_TYPE
XREF_SECURITY_ID
XREFERENCE_HIST
SECURITY_ALIAS
XREF_SECURITY_ALIAS
XREF_TYPE
XREF_SECURITY_ID
RATINGS
SECURITY_ALIAS
SRC_INTFC_INST
EFFECTIVE_DATE
SECURITY_ANALYTICS
SECURITY_ALIAS
SRC_INTFC_INST
EFFECTIVE_DATE
SECURITY_ANALYTICS_EQUITY
SECURITY_ALIAS
SRC_INTFC_INST
EFFECTIVE_DATE
SECURITY_ANALYTICS_FI
SECURITY_ALIAS
SRC_INTFC_INST
EFFECTIVE_DATE


RULES
ENTITY
ENTITY_ID
ENTITY_TYPE
ENTITY_EXTENSION
ENTITY_ID
ENTITY_CHARACTERISTICS
ENTITY_ID
CODE
CODE_VALUE
ENTITY_HIST
ENTITY_ID
EFFECTIVE_DATE
ENTITY_TYPE
ENTITY_EXTENSION_HIST
ENTITY_ID
EFFECTIVE_DATE
ENTITY_CHARACTERISTICS_HIST
ENTITY_ID
EFFECTIVE_DATE
CODE
CODE_VALUE
ENTITY_XREFERENCE
INSTANCE
ENTITY_ID
XREF_ACCOUNT_ID
XREF_ACCOUNT_ID_TYPE
ENTITY_DETAIL
ENTITY_ID
ENTITY_DETAIL_ID
ENTITY_TYPE
ENTITY_LIST
INSTANCE
ENTITY_ID
CODE_VALUE
ENTITY_DETAIL_HIST
ENTITY_ID
ENTITY_DETAIL_ID
EFFECTIVE_DATE
ENTITY_TYPE
CLIENT
INSTANCE
CLIENT_ADDRESS
INSTANCE
EFFECTIVE_DATE
ADDRESS_TYPE
CLIENT_INST
CLIENT_RELATIONSHIPS
INSTANCE
ENTITY_ID
CLIENT_INST
RELATIONSHIP_TYPE
EFFECTIVE_DATE
CONTACT_INST
CLIENT_HIST
INSTANCE
EFFECTIVE_DATE


HOLDING
POSITION
POSITION_ID
ENTITY_ID
EFFECTIVE_DATE
SRC_INTFC_INST
POSITION_INCOME_DETAIL
POSITION_ID
POSITION_DETAIL_ID
POSITION_DETAIL
POSITION_ID
POSITION_DETAIL_ID
SECURITY_ALIAS
POSITION_COST_DETAIL
POSITION_ID
POSITION_DETAIL_ID
POSITION_COST_DETAIL_USER
POSITION_ID
POSITION_DETAIL_ID
POSITION_INCOME_LOT
POSITION_ID
LOT_LEVEL_POSITION
LOT_LEVEL_POSITION
POSITION_ID
LOT_LEVEL_POSITION
SECURITY_ALIAS
POSITION_COST_LOT
POSITION_ID
POSITION_LOT_ID
POSITION_COST_LOT_USER
POSITION_ID
LOT_LEVEL_POSITION


PERFORM
PERF_SUMMARY
DICTIONARY_ID
PERF_SUM_INST
ENTITY_ID
SRC_INTFC_INST
PERF_FREQ_CODE
PERF_SEC_RETURNS
PERF_SUM_INST
SECURITY_ALIAS
PERF_ROLLUP_RETURNS_ID
ENTITY_ID
RULES
DICTIONARIES
DICTIONARY_ID
PACE_MASTER
CODES
INSTANCE
CODE_VALUES
INSTANCE
INTFC_INST
SHORT_DESC
DICTIONARY_DETAIL
DICTIONARY_ID
DICTIONARY_DETAIL_ID
DICTIONARY_ITEM_ID
DICT_L1_CODE_VALUE
DICT_L2_CODE_VALUE
DICT_L3_CODE_VALUE
DICT_L4_CODE_VALUE
DICT_L5_CODE_VALUE
DICTIONARY_FORMAT
DICTIONARY_ID
DICTIONARY_LEVEL
FIELD_ATTRIBUTE_ID
DICTIONARY_LEVEL_DESC
FIELD_ATTRIBUTES
FIELD_ATTRIBUTE_ID
REF_CODE_ID
DATABASE_NAME
TABLE_NAME
FIELD_NAME


CASH
TRADES
CASH_FLOW_PROJECTION
CASH_FLOW_INST
ENTITY_ID
PROJ_EFFECTIVE_DATE
EFFECTIVE_DATE
SRC_INTFC_INST
SECURITY_ALIAS
TRANS_NUM
CASH_ACTIVITY
CASH_INT
ENTITY_ID
EFFECTIVE_DATE
SRC_INTFC_INST
SECURITY_ALIAS
TRANS_NUM
TRADE
TRADE_ID
ENTITY_ID
EFFECTIVE_DATE
SRC_INTFC_INST
SECURITY_ALIAS
TRANSACTION_ID
TRADE_DETAIL
TRADE_ID
CORP_ACT_HIST
INSTANCE
SECURITY_ALIAS
TO_SECURITY_ALIAS
EFFECTIVE_DATE
SRC_INTFC_INST
TRANS_TYPE
CORP_ACT_ID
SWEEP_DATE
SECURITY
MUNI
SECURITY_ALIAS
FIXED_INCOME
SECURITY_ALIAS
SECURITY_MASTER
SECURITY_ALIAS
EQUITY
SECURITY_ALIAS
MBS
SECURITY_ALIAS
MUNI_HIST
SECURITY_ALIAS
SRC_INTFC_INST
EFFECTIVE_DATE
MUNI_DETAIL
SECURITY_ALIAS
MUNI_DETAIL_HIST
SECURITY_ALIAS
SRC_INTFC_INST
EFFECTIVE_DATE
FIXED_INCOME_HIST
SECURITY_ALIAS
SRC_INTFC_INST
EFFECTIVE_DATE
FIXED_INCOME_DETAIL
SECURITY_ALIAS
FIXED_INCOME_DETAIL_HIST
SECURITY_ALIAS
SRC_INTFC_INST
EFFECTIVE_DATE
SECMASTER_HISTORY
SECURITY_ALIAS
SRC_INTFC_INST
EFFECTIVE_DATE
SECURITY_MASTER_DETAIL
SECURITY_ALIAS
SECURITY_MASTER_DETAIL_HIST
SECURITY_ALIAS
SRC_INTFC_INST
EFFECTIVE_DATE
EQUITY_HIST
SECURITY_ALIAS
SRC_INTFC_INST
EFFECTIVE_DATE
EQUITY_DETAIL
SECURITY_ALIAS
EQUITY_DETAIL_HIST
SECURITY_ALIAS
SRC_INTFC_INST
EFFECTIVE_DATE
MBS_HIST
SECURITY_ALIAS
SRC_INTFC_INST
EFFECTIVE_DATE
MBS_DETAIL
SECURITY_ALIAS
MBS_DETAIL_HIST
SECURITY_ALIAS
SRC_INTFC_INST
EFFECTIVE_DATE