Perceptive Data Warehouse

Fundamentals

Version: 2.0.x

Written by: Product Knowledge, R&D
Date: November 2018

Hyland™

© 2018 Hyland Software, Inc. and its affiliates.

Table of Contents

ETL Process **4**
Table naming scheme **6**
Table Names 6
Table types and construction **7**
Fact tables 7
Dimension tables 7
Throughput tables 8
Trigger tables 8
Column name structure **8**

ETL Process

Data is sent in JSON format from the Perceptive Experience client to the data warehouse through the push service. JSON formatting is an open standard data format used to transmit data in a readable text string using key-value pairs. This JSON data must be extracted, transformed, and loaded into the warehouse through a number of steps to make it more consumable for queries and reports. This processing of data is commonly referred to by the acronym ETL for Extract, Transform, and Load.

As the data is extracted from the JSON, it is evaluated to determine if it falls into two different categories based on predefined algorithms.

- **Fact data** - does it provide information or metrics specific to a business process?
- **Dimension data** - does it provide descriptive or attribute information in support of a fact table or business process?

After those determinations have been made, the data is then transformed and cleansed to ensure the data is in the proper format or structure. There may be no transformation necessary, meaning that the data simply moves from the extract stage to the load stage with no transformation applied. For other portions of the data, there may need to be sorts, aggregates, lookups, or other transformations applied at this stage before it gets loaded into the warehouse.

The load process moves all the transformed data into the warehouse in the following manner.

- For fact tables, new rows are created for each new data set, including reference keys to necessary dimension tables.
- For dimension tables, data inserts a new row for each data set or updates an existing data set depending on the type of dimension (for example, regular versus a slowly changing dimension). This is further discussed in the Dimension tables section.
- For throughput tables, each row can be updated numerous times for multiple data sets. This is further discussed in the Throughput tables section.

The diagram below gives a high-level overview of the relation between a typical fact table and the respective dimension tables. Note that a reference key from the fact table refers to a primary key from one of the dimension tables.

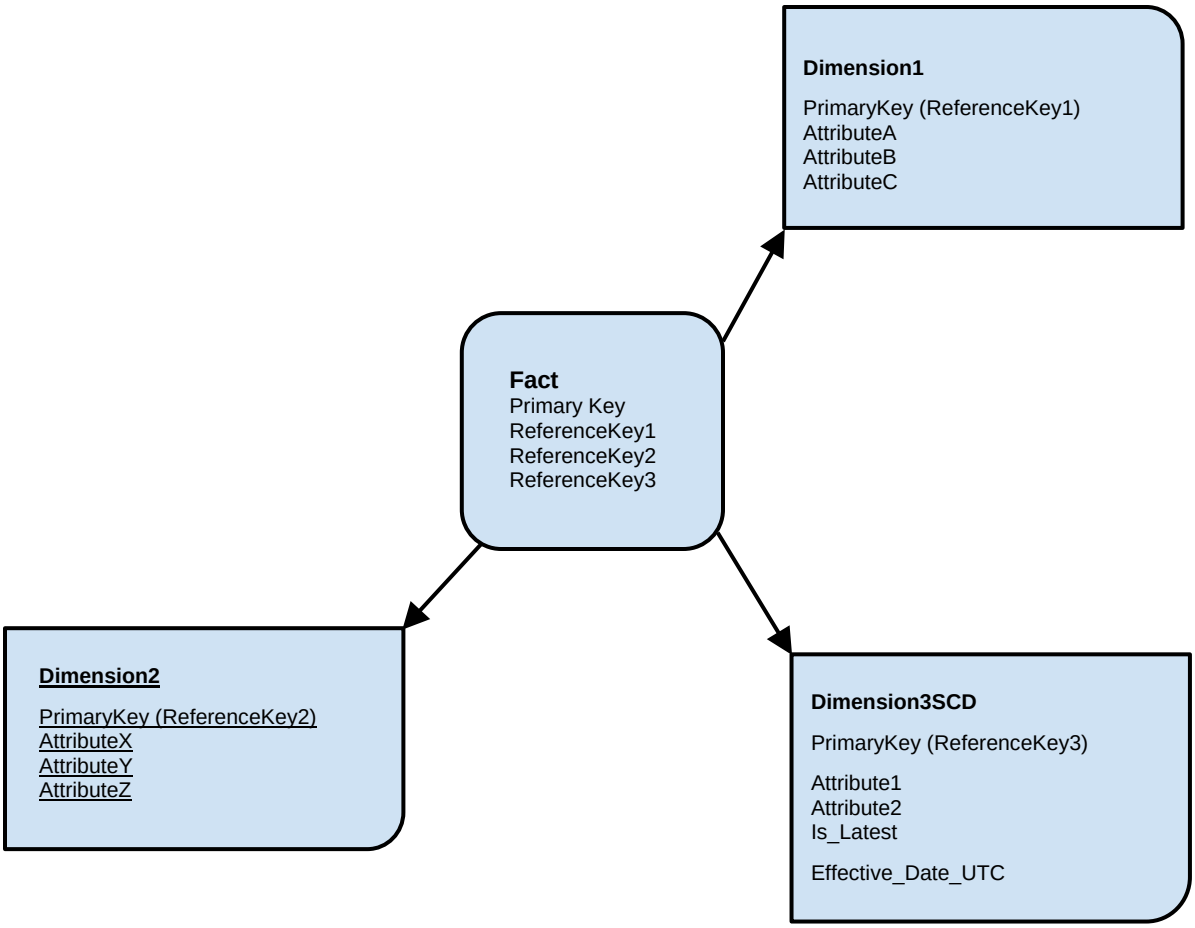
Dimension1
PrimaryKey (ReferenceKey1)
AttributeA
AttributeB
AttributeC
Fact
Primary Key
ReferenceKey1
ReferenceKey2
ReferenceKey3
Dimension2
PrimaryKey (ReferenceKey2)
AttributeX
AttributeY
AttributeZ
Dimension3SCD
PrimaryKey (ReferenceKey3)
Attribute1
Attribute2
Is_Latest
Effective_Date_UTC

Table naming scheme

To provide a consistent naming style to the data warehouse tables, the following naming convention is used when creating new tables. Understanding the naming scheme will assist you in understanding the purpose of the table and how it relates to other tables in the warehouse.

Table Names

There are five main groupings assigned to each table name that always follow this pattern: (Package)_(Solution)_(Object)_(Granularity)_(Star Schema Monicker).

Table Name Grouping	Purpose
Package	A package is a collection of solutions that may be branded for a particular product (such as CORE, HCCI).
Solution	Solutions are modular offerings based on a Perceptive Experience product delivered within a package (such as BC, RC).
Object	Objects categorize types of things (such as BATCH, DOC, PAGE, STEP, ORIGIN).
Granularity	Granularity explains the detail at which an object is broken down (such as USER_STEP, ACTION).
Star Schema Monicker	Monickers identify how the table fits into a typical star schema role. They can be Dimensions (D), Facts (F), Throughput (TP) or Trigger (TR) tables.

Table Name examples are provided in the following table.

Package	Solution	Object	Granularity	Star Schema	Table Name
Healthcare Capture and Indexing	Batch Capture	Document	Action	Fact	HCCI_BC_DOC_ACTION_F
Core	--	Date	--	Dimension	CORE_DATE_D
Healthcare Capture and Indexing	Batch Capture	Batch	Job	Trigger	HCCI_BC_BATCH_JOB_TR

Table types and construction

Fact tables

Fact tables are comprised of several reference keys that reference various dimension tables. For example, the HCCI_BC_BATCH_ACTION_F table has the following structure.

Column Name	Column Type	Table Reference
HCCI_BC_BATCH_D_KEY	primary key	--
HCCI_BC_BATCH_TYPE_D_KEY	reference key	HCCI_BC_BATCH_TYPE_D
CORE_USER_D_KEY	reference key	CORE_USER_D
HCCI_STEP_D_KEY	reference key	HCCI_STEP_D
HCCI_ORIGIN_D_KEY	reference key	HCCI_ORIGIN_D
HCCI_SOLUTION_D_KEY	reference key	HCCI_SOLUTION_D
HCCI_BC_ACTION_D_KEY	reference key	HCCI_BC_ACTION_D
CORE_DATE_D_KEY_UTC	reference key	CORE_DATE_D
CORE_TIME_D_KEY_UTC	reference key	CORE_TIME_D
CORE_DATE_D_KEY_LOCAL	reference key	CORE_DATE_D
CORE_TIME_D_KEY_LOCAL	reference key	CORE_TIME_D
ACTION_DATETIME_LOCAL	timestamp	--

Dimension tables

Where fact tables use reference keys, dimension tables use readable column names to provide the information on the data within. They can then be linked back to their associated fact tables to provide proper context. Dimension tables can be split into two categories: regular dimension tables and slowly changing dimensions (SCD).

Regular dimension tables, such as HCCI_BC_ACTION_D, CORE_DATE_D, are updated with new rows without making any updates or modifications to the old rows. Regular dimensions are comprised of a primary key, descriptive column names and a timestamp to indicate when the row was inserted into the warehouse.

Slowly changing dimension tables, such as HCCI_BC_BATCH_D, CORE_DOC_TYPE_D, receive updates to existing rows. However, to maintain historical integrity, the old rows remain intact and new rows with the new data are inserted. As with regular dimensions, a primary key and descriptive column names are present in the SCD with the addition of an IS_LATEST flag to identify if the row is the most recent or not. There are also an EFFECTIVE_DATE_UTC column, which is populated when the row is

initially added, and an EXPIRED_DATE_UTC column, which is updated when the row is updated to become out of date.

Throughput tables

A throughput table is a specialized warehouse table that provides aggregated values for the consumer. Since they are pre-aggregated views of data, they are not as flexible as the rest of the warehouse, but they do provide tremendous value to produce quick answers for specific questions.

Throughput tables, like the Fact and Dimension tables, have a primary key associated with them. They also reference keys from other dimension tables, similar to fact tables. The unique type of field you typically see in a throughput table is a metric such as a count (such as PAGE_COUNT) or a time interval (such as BATCH_DURATION_SECONDS), which is the aggregated value.

One row of a throughput table may get updated multiple times during the course of a Perceptive Experience process. It is for this reason that some fields in a throughput table may point to a key value indicating it is “Not Found” when in reality it simply has yet not been set. For example, the HCCI_BC_BATCH_TP table contains a field named STOP_CORE_DATE_D_KEY_UTC. If an item has yet to be processed out of a Step, this field would have a key value of 2. This references the CORE_DATE_D.CORE_DATE_D_KEY value of 2, indicating it is Not Found or not set. After the item has been processed out of the Step and the job has been run, the field would now have a key value like 20150806 referencing the date when it moved out of the Step.

Trigger tables

Trigger tables are an interim storage table used for the processing of data. After a job is run, the data clears out of this table. In the event there is an error processing the data, it remains in this table with the ERROR_DURING_PROCESSING flag set to 1. You can reset this to 0 and rerun the job to process the data.

Column name structure

The naming structure for column or field names falls into three main categories: primary keys, reference keys, and typical columns.

A Primary Key is the main key for the table to which it belongs. There is only one primary key per table and all tables contain a primary key. The naming convention for primary keys is relatively simple: (table name)_Key.

Example: `HCCI_BC_ACTION_D_KEY`

References Keys are keys that refer to other tables within the schema. Depending on the table, there may be no reference keys or there may be multiple reference keys. The naming convention for references keys is: (Optional Prefix)_(Key Name)_(Optional Suffix).

Example: The Start of an action, Time Key, Local Time
`START_CORE_TIME_D_KEY_LOCAL`

Note Date/Time Suffixes will be either `LOCAL` or `UTC`.

Typical column names refer to any column that is not a primary or reference key. This could be a metric from a throughput table or a name field from a dimension. The naming convention for typical column names is: (Object)_(Attribute)

Example: `AC`