

Modeling master data

Design patterns map out entity relationships to improve accuracy. *by Taher F. Borsadwala*

Businesses must understand their data to successfully use it. According to Jill Dyché and Evan Levy in their book, “Customer Data Integration,” a business usually has four main types of data: transactional, reference, relationship and metadata. Reference data uniquely identifies a product, customer or other business entity. For instance, a customer’s first and last name both represent reference data. Relationship data describes how an entity relates to other entities.

Master data comprises reference and relationship data and enables companies by providing information about business events and transactions. Its unique responsibility and what sets it apart from all other forms of data are the details it provides, such as a description of the product, the components of the product materials and the name of the customer who purchased the product.

Master data also identifies the business event’s structure by providing a map of the references, relationships, hierarchies and matrices. This map, called a master data design pattern, aggregates and summarizes the relationships among entities as they appear in a semantic layer and in physical storage. Furthermore, the details provided by master data design patterns significantly affect the performance and accuracy of the semantic layer and overall total cost of ownership of the data warehouse environment.

Master data modeling design patterns

In many enterprise data models (EDMs), the categories are industry- and implementation-

Pattern name	Description
Relationship Design Pattern	Within a single table, an entity has a relationship to other entities at any given point in time.
X-Ref Design Pattern	An entity maps to a standard entity at any given point in time.
X-Ref and Relationship Design Pattern	An entity maps to a standard entity and has a relationship to other entities in a single table at any given point in time.

specific, whereas the master data is abstract and not clearly segregated. This enables the master data design pattern models to be used across multiple industries and for multiple purposes. The EDM, meanwhile, contains placeholders for the references and relationships portion of the master data, so the specific information can be dropped into those slots once they are defined in one of the design patterns.

Placeholders are not provided when a data structure simply exists to cater to master data. The design pattern data models can cater to multiple entities and are not confined to just one at a time. If such data models are created as part of the EDMs, any industry- or implementation-specific data should be merged with the enterprise data to enable master data to have its own demarcation. This demarcation of the master data will help simplify data management.

EDMs support master data on an as-needed basis. For instance, in the case of mergers or acquisitions, an organization table is needed in the master data design models to show hierarchical relationships. In such a

scenario, a recursive relationship is created by having a parent org_id in the Organization table, making relationships in the master data industry- and implementation-specific. A similar approach might be used when hierarchies are created on other tables within the EDM.

Designing an appropriate data structure for master data requires the selection of a design pattern that supports how instances of master data are related. Because the enterprise data warehouse (EDW) provides placeholders for master data, a strong tie between the EDW and master data models is necessary. Naturally, not every eventual use of data can be anticipated, so in general, reusable and normalized master data structures remain the most economical choice.

A master data modeling design pattern is, basically, a template of reusable solutions that can be applied to commonly occurring problems or scenarios in many different situations. Three general types of master data design patterns exist: Relationship Design Pattern, X-Ref Design Pattern, and X-Ref and

Relationship Design Pattern. (See table 1, page 1.)

Each design pattern is unique and is identifiable by the following traits:

- > **Pattern name.** A descriptive and unique name used to identify the pattern
- > **Intent.** The purpose and goal of the pattern
- > **Applicability.** The situation and context in which the pattern is used
- > **Structure.** A graphical representation of the pattern
- > **Scenario.** An illustration of how the pattern can be used
- > **Implementation.** A description of the solution implemented by the pattern

These features identify each pattern's specifications for modeling master data. A breakdown of how these features apply to the Relationship Design Pattern follows.

Relationship Design Pattern

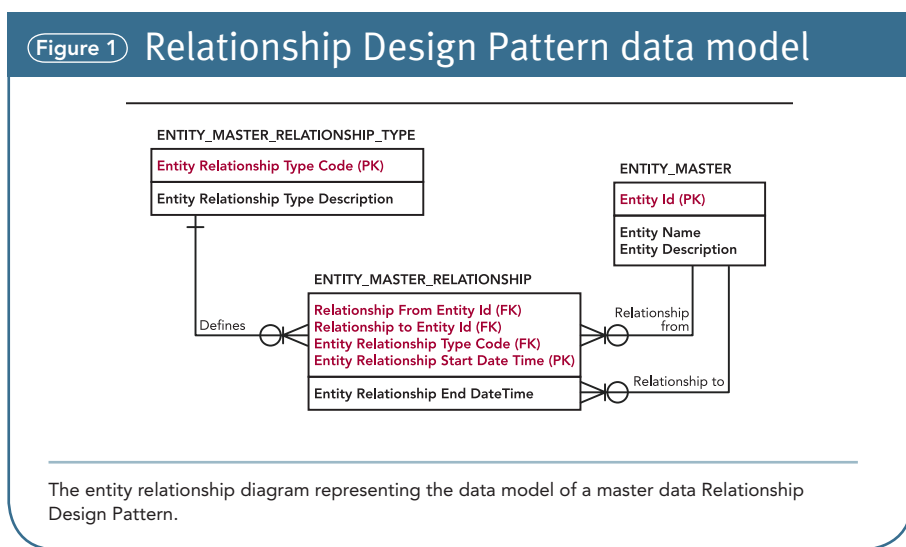
Parent-child or business relationships that are time-bound or time-dependent are defined in the Relationship Design Pattern. An example of a parent-child relationship that is time-bound is a sales division that belongs to one business unit today but that might belong to another business unit tomorrow. An example of a product that has a time-dependent business relationship is an MP3 player that has become more sophisticated over time. The following features identify the Relationship Design Pattern.

INTENT

The relationships among entities can vary at different points in time.

APPLICABILITY

This pattern can cater to hierarchical relationships (parent-child), business relationships among entities, entities in hierarchical relationships having business relationships with other entities, linear relationships among entities, and relationships between alternates or substitutes. An alternate or a substitute is an entity that can be used in



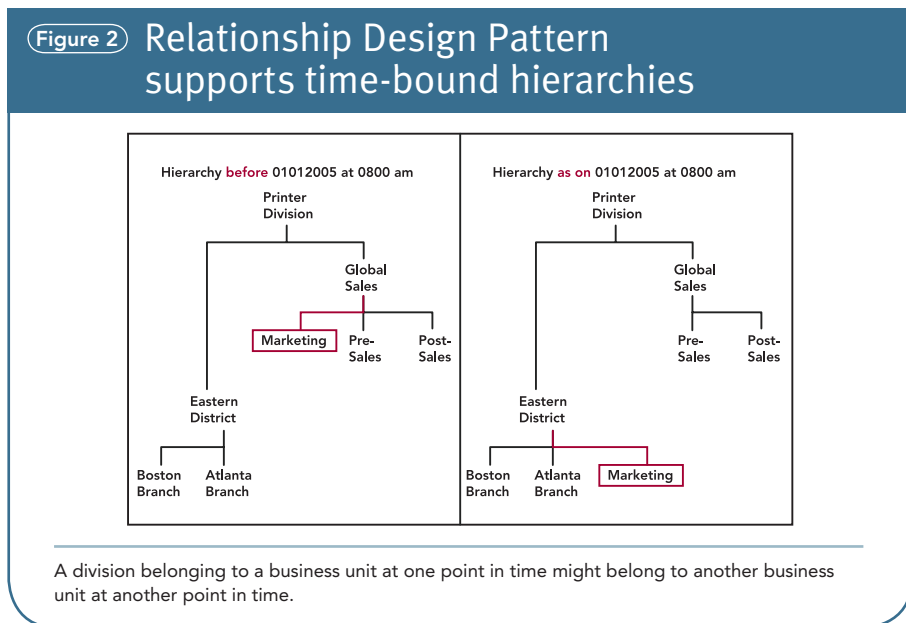
place of another. For example, a car could have leather, plush or plain seats. These three stand-alone products are substitutes/alternates of one another.

STRUCTURE

Data stored in a master table is used by corresponding tables to identify relationships between entities. (See figure 1.) The following tables make up the structure:

- > **ENTITY_MASTER** contains the corporate records, sometimes referred

to as “golden records.” These records contain master data, which is core to the architectures of individual applications and provide consistency and integration across applications. Codes, names and identifiers must be well defined in terms of their business meaning, data model and appropriate application use. In all, these are considered the central, trusted source of data tables since applications access the one master table directly.



Party Id	Party Name
100	Atlanta Branch
101	SE Branch
102	Southern Div
103	Peachtree City

- > **ENTITY_MASTER_RELATIONSHIP** is used for mapping the child entity to the parent entity and various parent-child or other business relationships.
- > **ENTITY_MASTER_RELATIONSHIP_TYPE** holds different hierarchical relationship types, such as parent-child or business relationships.

SCENARIO

Figure 2 (page 2) shows an example of how the Relationship Design Pattern supports time-bound hierarchies using an enterprise table and an organization table. In this case, the ENTITY_MASTER holds data pertaining to the various organizations. To map it to the scenario provided, Printer Division, Eastern District, Global Sales, Boston Branch, etc., are stored in this table.

In the table ENTITY_MASTER_RELATIONSHIP, Printer Division is stored as the From entity (parent) while Eastern District is stored as the To entity (child).

Branches in the ENTITY_MASTER_RELATIONSHIP_TYPE show the relationship between Eastern District and Boston Branch, while Sales identifies the relationship between Global Sales and Pre-Sales.

Hierarchy nodes (child) can belong to different parents at different times. For instance, Marketing (child) belonged to the business unit Global Sales (parent) before Jan. 1, 2005, but to the business unit Eastern District (parent) as of Jan. 1, 2005.

IMPLEMENTATION

To improve data access performance, these indexes can be applied: unique primary

index (UPI), non-unique primary index (NUPI), unique secondary index (USI) and non-unique secondary index (NUSI), and multilevel partitioned primary index (MLPPI) if the start and the end dates repeat. For time-bound data and data that is specifically range-based, employing partitioned primary indexes (PPIs) can make the pattern highly efficient.

To weigh the benefits of applying the suggested indexes on the actual table structures that use the pattern, review the resulting EXPLAIN text.

X-Ref Design Pattern

The X-Ref Design Pattern defines cross-references among source and target entities. A golden record can be introduced using this pattern. This pattern can be used when multiple entities mean the same but one of those entities is denoted as the golden or corporate record.

INTENT

Multiple source system entities map to a common pool of target master entities; within that, an entity has a relationship to a standard entity and is time-dependent.

APPLICABILITY

Duplicate entities are related to a single

master entity or map local entities to corporate entities.

STRUCTURE

Relationships can evolve in three ways:

- > Introducing an ENTITY_MASTER table in the EDW to hold the golden records
- > Handling the golden records within the ENTITY table without introducing the ENTITY_MASTER table
- > Creating a common target table that holds all source attributes

Figure 3 shows how relationships exist in an ENTITY table. ENTITY contains the local source records. To fulfill this pattern's intent, these tables need to be introduced:

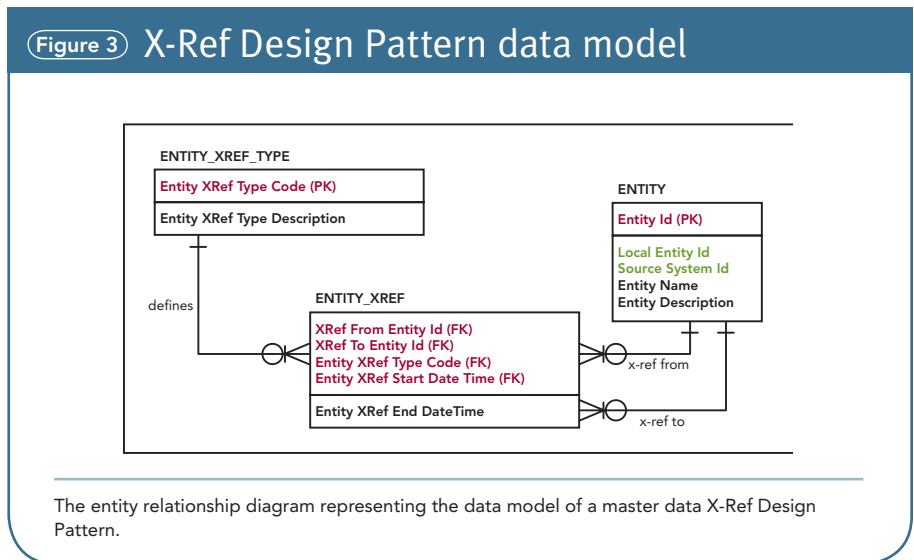
- > **ENTITY_XREF**. Holds the ENTITY cross-reference mappings to a golden or corporate record.
- > **ENTITY_XREF_TYPE**. Contains the different cross-reference relationships.

SCENARIO

In table 2 on page 3, all of the listings in the Party Name column are duplicates, the most accurate of which is Atlanta Branch.

PARTY_XREF_TYPE (table 3) contains the Party Gold Master relationship type that will map duplicate PARTY records with a single PARTY_MASTER golden record.

PARTY_XREF (table 4) contains the



Party XRef Type Code	Party XRef Type Description
1	Party Gold Master

cross-reference relationships found in the golden record that resides in the PARTY table, and checks for duplicate records.

IMPLEMENTATION

As in the Relationship Design Pattern described above, the indexes UPI, NUPI, USI and NUSI can be applied to improve data access performance. Review the EXPLAIN text to determine the benefits of applying these indexes on the table structures.

X-Ref and Relationship Design Pattern

Cross-references are first defined among source and target entities, then parent-child relationships are defined among the various

targets. Matrix creations, such as a cube representation of time-bound cross-references and relationships, are an example of when this pattern would be used.

INTENT

An entity has a cross-reference relationship to a standard entity, and that standard entity has a parent-child relationship with another standard entity. These relationships coexist and are time-dependent. (See figure 4.)

APPLICABILITY

This pattern is applicable to hierarchical relationships that cross-reference to duplicates.

STRUCTURE

ENTITY contains the local/source records. To fulfill this pattern's intent, the following tables need to be introduced:

XRef From Party Id	XRef To Party Id	Party XRef Type Code	Party XRef Start DateTime
100	101	1	09012006 4:31 pm
100	102	1	12012007 2:00 am
100	103	1	07042007 12:00 pm

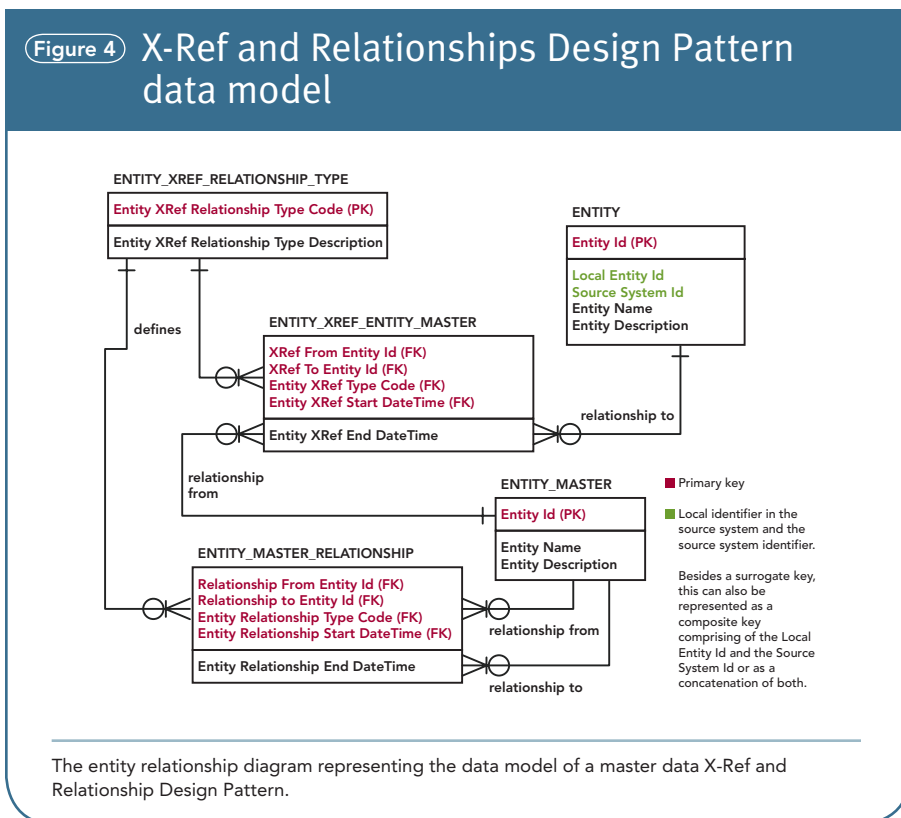
- > ENTITY_MASTER houses the golden or corporate record.
- > ENTITY_MASTER_RELATIONSHIP maps child entity to the parent entity.
- > ENTITY_MASTER_RELATIONSHIP_TYPE identifies the different hierarchical relationship types.
- > ENTITY_XREF_ENTITY_MASTER contains the cross-reference mapping between the ENTITY and the ENTITY_MASTER.
- > ENTITY_XREF_RELATIONSHIP_TYPE carries the different cross-reference relationships.

SCENARIO

Time-bound matrices or cubes are supported in this design pattern. Nodes can be the golden records that are cross-referenced to the duplicates. Such duplicates indirectly become child nodes to the parent node of the golden record. For instance, because the golden record Atlanta Branch is a child of Eastern District, SE Branch (referencing golden record Atlanta Branch) also becomes a child of Eastern District by default. (See figure 5, page 5.)

IMPLEMENTATION

As the name suggests, this design pattern data model is a combination of the Relationship Design Pattern and the X-Ref Design Pattern. With the functions of both patterns, this hybrid data model can cater to cube and matrix structures, and also makes use of the various indexes (UPI, NUPI, USI, NUSI, PPI AND MLPPi) to improve performance. However, the EXPLAIN text should be



reviewed to weigh the benefits of these indexes on the actual table structures.

Know your data

Data model design patterns help maintain structure and order in an organization’s data. Applying a map that links data to each entity and business relationship within an event or transaction enables companies to know the characteristics of its data and thus make better decisions based on that knowledge. **T**

Taher F. Borsadwala, a Teradata Master, is a Solution Lead in the Teradata Master Data Management Center of Excellence.

Online For more on Teradata Master Data Management, visit Teradata.com/master-data-management.

