Understanding the PMT Data Model

The goal of this paper is to explain the basic PMT data model concepts, using easy to understand examples. The focus of this paper is to explore Taxonomy, the main feature of the PMT data model that makes PMT highly scalable and flexible.

Draft Version

Spatial Development International, LLC

2208 NW Market Street, Suite 202
Seattle, WA 98107
http://www.spatialdev.com

Understanding the PMT Data Model

What is Taxonomy?

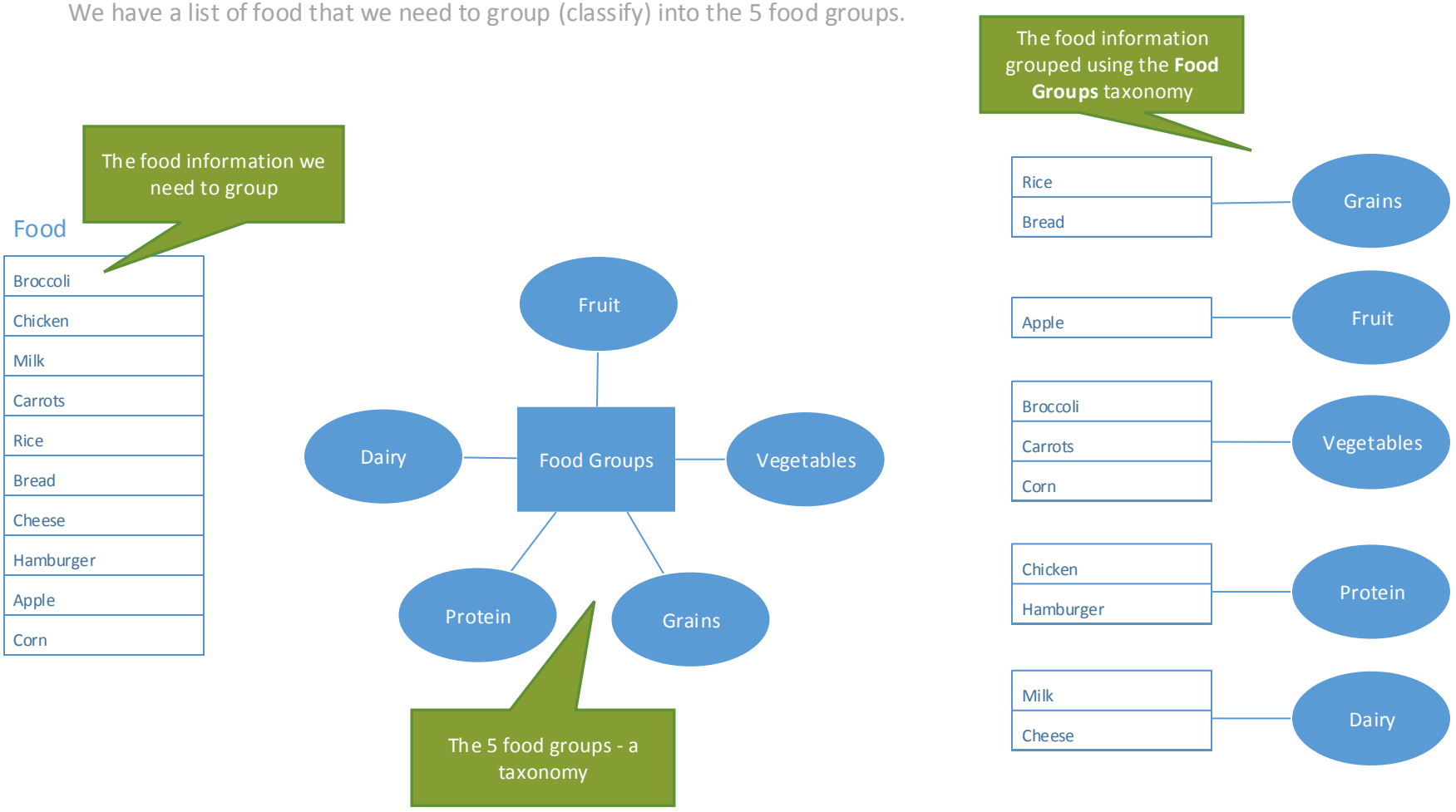
Taxonomy is how we classify information.

What does it mean to classify information?

To classify, means to group information into categories.

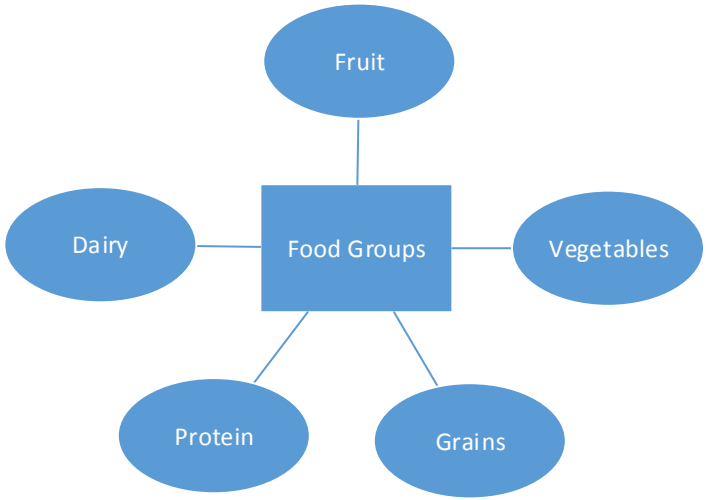
An example:

We have a list of food that we need to group (classify) into the 5 food groups.



Let's take a closer look at Taxonomy.

Taxonomy has two main elements. A taxonomy has a name and one or more classifications. Lets look closer at our Food Groups taxonomy example.



The taxonomy name is:

Food Groups

The taxonomy classifications are:

- Fruit
- Vegetables
- Dairy
- Protein
- Grains

What does taxonomy look like in the database?

Like all data in the database, taxonomy is stored in tables. Taxonomy is stored using **TWO** tables: the taxonomy table and the classification table.

Taxonomy Table

taxonomy_id	name
132	Food Groups

The taxonomy name goes into the taxonomy table. Each taxonomy is given a unique number called a primary key.

Classification Table

classification_id	taxonomy_id	name
433	132	Fruit
434	132	Vegetables
435	132	Dairy
436	132	Protein
437	132	Grains

The taxonomy classifications go into the classification table. Each classification is given a unique number (primary key) and has the taxonomy_id that contains its taxonomy name.

Understanding the PMT Data Model

Let's take a closer look at how we store information in the PMT database.

All information in the database is stored in tables. In the PMT database there are **THREE** different kinds of tables: taxonomy tables, entity tables and junction tables.

Taxonomy tables.

Taxonomy has TWO tables: taxonomy and classification. These contain all the taxonomy data for the database.

Entity tables.

Tables that contain distinct and independent information. Entity tables have data that, all by itself, has meaning.

Junction tables.

Tables that create a relationship between two or more Entity tables. A junction table, all by itself, has **NO** meaning.

Next, let's look at what our food information looks like in the database.

The food information is stored in a table with a primary key. The food table is an entity table, as its information holds a distinct meaning, independent from other information in the database.

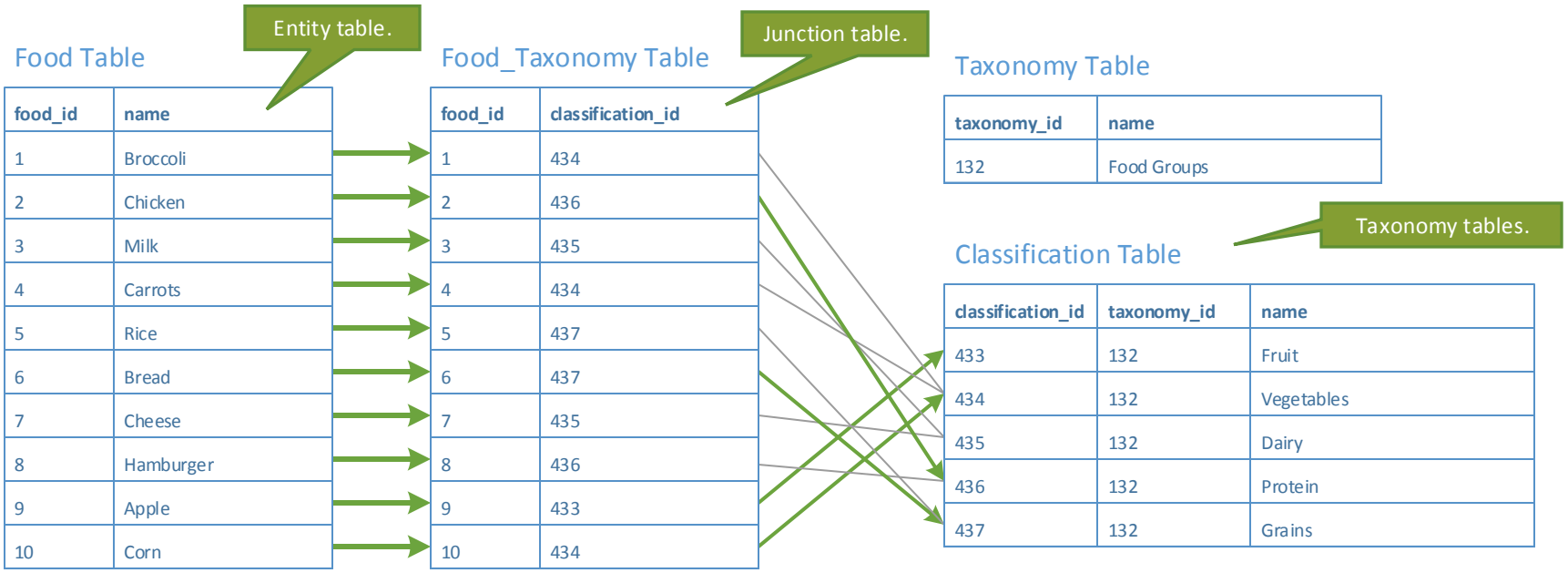
Food Table

food_id	name
1	Broccoli
2	Chicken
3	Milk
4	Carrots
5	Rice
6	Bread
7	Cheese
8	Hamburger
9	Apple
10	Corn

Understanding the PMT Data Model

How is taxonomy used in the database?

Given the food table and taxonomy tables, we can add a junction table, so that we can create a relationship between our food information and our Food Group taxonomy.



Food Table

food_id	name
1	Broccoli
2	Chicken
3	Milk
4	Carrots
5	Rice
6	Bread
7	Cheese
8	Hamburger
9	Apple
10	Corn

Food_Taxonomy Table

food_id	classification_id
1	434
2	436
3	435
4	434
5	437
6	437
7	435
8	436
9	433
10	434

Taxonomy Table

taxonomy_id	name
132	Food Groups

Classification Table

classification_id	taxonomy_id	name
433	132	Fruit
434	132	Vegetables
435	132	Dairy
436	132	Protein
437	132	Grains

How would this data be stored traditionally?

Traditionally, a flat database structure would store the food group information directly in the table. In a normalized database structure, a foreign key in the food table would directly tie a food to a food group in the food group table.

Food Table

food_id	name	food_group
1	Broccoli	Vegetable
2	Chicken	Protein
3	Milk	Dairy
…	…	…

Flat Data Structure

Food Table

food_id	group_id	name
1	23	Broccoli
2	24	Chicken
3	25	Milk
…	…	…

Food Group Table

group_id	food_group
23	Vegetable
24	Protein
25	Dairy
…	…

Normalized Data Structure

Why does PMT use taxonomy and why is it so **POWERFUL**?

PMT uses taxonomy so that it is highly scalable and flexible, making it a powerful data model. Remember that taxonomy allows us to group information in the database. Unlike the traditional approaches to grouping and classifying information, taxonomy allows for the **ADDITION** of an infinite number of classifications, **WITHOUT** changing the data model.

Databases are rarely stand alone. More than not, databases are the source for information for many (even hundreds or thousands of) applications, documents and processes. Any change to the data structure has the potential to impact all of these dependences. As our businesses change and grow, the database supporting the business information needs to be just as flexible.

Let's look at an example of this powerful feature.

Continuing with our food example, what would happen in the traditional data structures if we needed to group our food information by a **NEW** classification: **Food Source (*Animal* or *Plant*)**

The new classification in the Flat Data Structure

If we needed to group our food information by a **NEW** classification in the flat data structure, we are required to add a new column, changing the data structure of this table. For any application, document, or processes using this table, changes to those dependencies are **REQUIRED,** in order for them consume the new information.

Added a new column

Food Table

food_id	name	food_group	food_source
1	Broccoli	Vegetable	Plant
2	Chicken	Protein	Animal
3	Milk	Dairy	Animal
…	…	…	…

Flat Data Structure

The new classification in the Normalized Data Structure

Adding a **NEW** classification in the traditional normalized data structure requires a new column **AND** a new table, further complicating the **REQUIRED** updates to dependent applications, documents and processes

Added a new column

Food Table

food_id	group_id	source_id	name
1	23	53	Broccoli
2	24	54	Chicken
3	25	54	Milk
…	…	…	…

Food Group Table

group_id	food_group
23	Vegetable
24	Protein
25	Dairy
…	…

Food Source Table

source_id	food_source
53	Plant
54	Animal

New table

Normalized Data Structure

The new classification in the PMT Data Structure

Adding a **NEW** classification to the PMT data structure is not only **EASY**, but it requires **ABSOLUTELY NO DATA STRUCTURE CHANGES**! Application, documents and process can consume the new information **AUTOMATICALLY**. Let's see what it looks like using Taxonomy in the PMT data structure:

Taxonomy Table

taxonomy_id	name
132	Food Groups
133	Food Sources

STEP #1
The name of the taxonomy "Food Sources" is added to the taxonomy table.

STEP #2
Add our classifications, linking them to the taxonomy name by its taxonomy_id.

Classification Table

classification_id	taxonomy_id	name
433	132	Fruit
434	132	Vegetables
435	132	Dairy
436	132	Protein
437	132	Grains
438	133	Plant
439	133	Animal

Nothing happens here.

Food Table

food_id	name
1	Broccoli
2	Chicken
3	Milk
…	…

Food_Taxonomy Table

food_id	classification_id
…	…
1	438
2	439
3	439
…	…

STEP #3
Add the relationships to the junction table

Other benefits of the PMT Data Model

Taxonomy in the PMT data model not only provides a data model that is scalable and flexible it also provides a normalized data structure reducing redundancy, promoting consistency and data integrity and allowing for closer model of real world entities processes and relationships.