# Basic Data Modeling Concepts

**Gerhard Beck**



For an overview of data modeling, see http://www.islandnet.com/~tmc/html/articles/datamodl.htm

There are actually two types of inter-related data models:[1]

**Logical Model** – Business view of the data which strictly follows the rules of normalization.

**Physical Model** – Physical view of the data. Rules of normalization may be relaxed somewhat for efficiency.

The basic steps in data modeling are:

1. Identify the things you are interested in. These are called ***entities*** in the logical model and ***tables*** in the physical model.

2. Identify ***relationships*** between entities. These become the lines drawn between each entity. For example, a manager has workers, workers are assigned workflows, and work flows are supported by tools. Note that managers are not directly related to tools, but are indirectly related through workers and work flows. Relationships can be either one or many. For example, a manager can have many workers, but each worker only has one manager.

3. Identify the ***keys*** for each entity. Keys are used to look up a row of data. Keys describe the minimum amount of data necessary to identify a particular thing. Often, a computer generated number is used for the key. For example, an employee ID number (generated by the HR software) might be used to identify each manager and each worker.

4. Identify the ***attributes*** for each entity. Attributes are the information which needs to be kept about each entity. For example, an worker's attributes might include the worker's name and e-mail address.

Data models are expressed in an ***Entity Relationship Diagram (ERD)*** and a ***Data Element Dictionary (DED)***. The ERD is a schematic representation of the database while the DED is a text representation. The two must be combined to get a clear picture of the data model.

[1] For a discussion of the differences between the two, see: http://www.dmreview.com/whitepaper/dmd.pdf

# Entity Relationship Diagram (ERD)

In the ERD, tables are represented as a pair of stacked boxes. The SQL name for the table is just above the box. The physical database key for the table is listed in the top box. The attributes (columns) of the table are listed in the bottom box.

The “symbology” of the table links is:

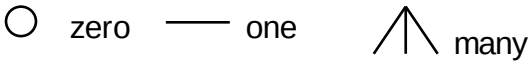
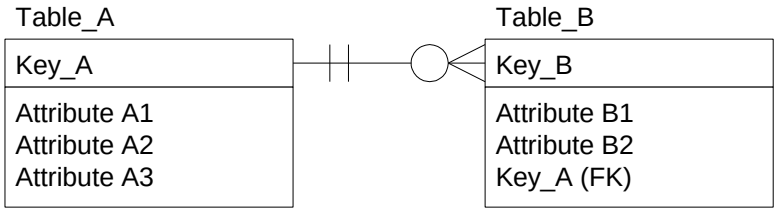


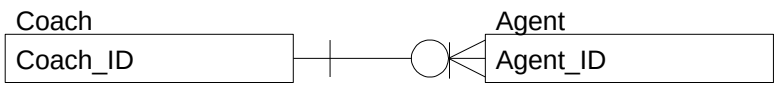
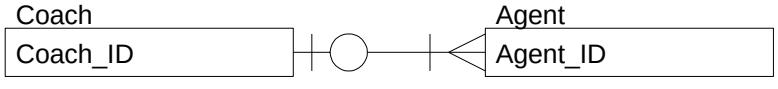

Table links are read by looking at one table then looking at the link attached to the second table. For example:




Here, each row in Table_A can have zero to many connections to Table_B. Each row in Table_B must have one and only 1 connection to a row in Table_A. An example of this type of relationship might be managers and workers. A new manager might not be assigned any workers. However, managers are typically assigned several workers. Each worker must have one and only one manager.

The actual column which connects the tables is identified by finding a common key column between the two tables. In this case, the common key column is Key_A. The (FK) behind Key_A in Table_B indicates that Key_A is a “Foreign Key”. A Foreign Key is used to look up a unique row in another table, in this case Table_A.

Here are some sample relationships with English translations:

| | |
|---|---|
|  | A manager may be associated with any number of worker.<br><br>An worker is associated with one and only one manager.<br><br>Implication: A manager must exist before an worker can be added. |
|  | A manager must be associated with one or more workers.<br><br>An worker may be optionally associated with a single manager.<br><br>Implication: An worker must exist before a manager can be added. |

One to many relationships can be either identifying or non-identifying.

***Identifying*** relationships are designated with a solid line and mean that if a row in the master table is deleted, then the row in the child table must be deleted. A corollary to this rule is that the key of the master table must be part of the key in the child table.

***Non-identifying*** relationships are designated with a broken line. A non- identifying relationship means that if a row in the one table is deleted, the corresponding row of the detail row need not be deleted. A corollary to that rule is that non- identifying relationships are attributes, not keys.

Physical data models sometimes aggregate a set of table where each of the tables consist of a one-up numeric key and a tag/name/label. These table typically drive pick lists in the application. The aggregated tables are described in a master “pick list table” while the items in each list are in the “pick item table”. Since the numbers are used to store references to the labels, the text of the label can be change without affecting the data stored in other tables. In other words, each place the tag is used will automatically get the new name associated with the number.