Data Modelling - Building the Data Dictionary
Design Issues: Attributes into Fields

Aim

Look at the design aspects of developing database Field Definitions from conceptual Attributes.

1.   Review Analysis issues

Some documentation of the results of top-down analysis is necessary. The following is a <u>guide</u> to what **may** have been established for entities. Such documentation must be appended to the ERM so that the results of analysis are clear and complete.

Entity name	a unique (agreed) name for the entity
Entity ID [optional]	a short reference name or code for the entity
Average/expected number of occurrences [optional]	an estimate of the number of occurrences (records) of the entity
Description	a definitive description of the entity [and, optionally, why it is included in the data model]
Synonym(s)	a list of other possible names for the entity
Relationship Link Phrases	**note that these do not have to be documented if they are already on the LDS**
Attributes	a list of attribute names

2.   Design aspects of developing field definitions for attributes

When converting attributes into fields, there are two areas of investigation and activity: (1) the definition of the field itself including all parameters necessary for applying the field to a database; (2) the usage of the field – how it is created, read or referenced, updated, and eventually deleted.

Field definition
Each attribute has associated with it a set or range of values it may take. This will specify some of the properties of the implemented field definition:
- data type
- size
- possible range of values
- possible set of values
- format
- units of measurement

Business Rules
Organisations have specific rules about the way that a system works, and the type of data that can be used in it. A number of business rules can be implemented when setting up database tables. More complex ones may need to be "programmed" into the system. Methods used include the following:

- Specifying the data type and/or field size
- Using input masks to specify the type of characters expected at each position in a string
- Using formats to specify how to display data
- Using field validation rules to limit the range of values allowed
- Specifying whether data is required (must be present)
- Specifying whether duplicate values are allowed in the table
- Defining relationships between tables to ensure referential integrity

Data Modelling - Building the Data Dictionary

The following need to be assessed and documented for <u>every</u> attribute of each entity. They should be documented on an entity-by-entity basis, and can be recorded on a data dictionary or a grid in preparation for implementing in a DBMS.

Attribute name	a unique (agreed) name for the attribute
Attribute ID [optional]	a short reference name or code for the attribute
Description	a definitive description of the attribute [and, optionally, why it is included in the data model]
Synonym(s)	a list of other possible names for the attribute
Data type	for example: alpha, alpha-numeric, numeric-integer, numeric-real, date/time, logical…
Size/length of field	in bytes – if the length can vary, then the maximum
Possible data values	any known pattern or range of possible real world values
Optionality	whether the field must always have a value or whether it can be null (note that fields used for keys should **never** be null)
Validation	any rules that can be applied to the field when a value is inserted; this applies mostly to codes which may have a known structure or to fields with a known range of values (such as a Grouped Domain)
Key usage	whether it is (part of) a Primary key or a Foreign key

Field use

The main consideration is how the values in the field are created, read or referenced, updated, or deleted (summarised in the acronym, CRUD!) How will the data values actually arrive in a field, and how they may subsequently be modified and later removed. The means by which values are inserted or changed are: data entry, processing, and/or house-keeping. The main concern is with requirements for data entry and processing.