Data Dictionary and Normalization

Priya Janakiraman

About Technowave, Inc.

Technowave is a strategic and technical consulting group focused on bringing processes and technology into line with organizational goals. Technowave provides best of breed solutions to real world problems for the Small to Medium Business (SMB) market. In addition to strategic consulting, Technowave provides system architecture, project management, development and implementation services. Technowave also provides technical and knowledgeable professional with Innovative Technologies to organizations on a flexible, contract and permanent basis focusing according to the client's dynamic needs.

For additional copies of this whitepaper, or to explore partnership opportunities with Technowave, please contact us as follows.

Technowave Inc.
900 Pump Road, Suite 69
Richmond , VA   23238
Tel: (804) 740-0957
Fax: (804) 740-0958

To achieve business excellence, it is important to increase sharing, integration and reusability of data. This paper explains what Data Dictionary is and how it helps us to increase consistency, clarity, completeness, ease of use, and reusability of data.

Data Dictionary is a collection of data describing the content, source, definition, structure, and business and derivation rules regarding the data within an organization. It is also called Metadata. Metadata is "data about the data", with examples being data types, lengths, scales, descriptions of the data elements and tables, entities, relationships, etc. Metadata data is stored in a repository to facilitate its accessibility. The metadata repository acts as a source of data to IT professionals in much the same manner as the Data Warehouse acts as a source of data to the business units within an organization.

For each database, Data Dictionary should contain the following information:

- Entity Types: The full name of the entity type and a description of the business purpose
- Attributes: Define the primary key attributes of the entity type. Each primary key attribute confirms a business definition and a description of why this column ensures an identity of the entity instance
- Relationships: The relation ships between entity types need to be drawn and in most cases the foreign keys need to be propagated, especially when foreign keys make up the primary key. Relationships show at least one descriptor, preferably in the parent to child direction
- Data Element Definition: Name of each filed and its properties. It includes data element number, element name, security classification, related data elements, data types, null value allowed or not, default value, validation rules, foreign key references and related reference documents

- Table definition: Name of each table and properties. It includes table owner, list of columns, order of columns, information about indexes, table organization, data element list and security classification
- Conceptual Data Model/Specification: Graphical representation of the information requirements of a business area
- View spec sheets and View diagrams
- Design analysis notes/samples and Logical Data Model
- DBA database space estimates: Number of tables per database and gross space estimate per database
- Load Timing: Gives required load frequency, and any other data loads. In addition, the expected error rate should be collected as part of the Operational Meta-Data
- List of potential tables that will be added during the project
- List of existing tables that will potentially be changed by the project
- A description of any uncertainty or risk in the project, because of the data source, etc

The following are the advantages of using a Data Dictionary:

Consistency: Corporate data, repositories, etc. are only successful when they are consistently accessed and maintained within an organization, especially as that data crosses organizational boundaries. Data Dictionary helps to maintain the consistency of corporate data across organizations.

Clarity: Data Dictionary makes data clear and usable for the business user and the developer. This supports efficient and consistent use of the data by both the originators and the various users of the data regardless of what divisional organization they belong to. Often, non-standardized data is used because data elements are known within the originating organization without regard to other users outside their organization. The lack of clarity can cause an outside user to misunderstand the meaning, use, or domain of a data element and so, create an erroneous report affecting a management decision.

Reusability: Data Dictionary support consistency which is a key ingredient in the ability of one divisional organization to incorporate work that has already been designed, tested, and approved by the corporation for reuse into their own new development projects. Reinventing the wheel costs money and time. Reusability is enabled by application of standards to produce consistent parts for fitting into future work.

Completeness: Data Dictionary helps an analyst know when data is clear, complete, and defined by specifying what completeness means and the steps to develop a complete data structure. Incomplete data properties or descriptions tend to be improperly used and misunderstand of data. They can also cost extra time for a developer to make multiple phone calls to clarify and complete the information needed to use the data.

Ease of Use for the Developer: Having clear and complete definitions/descriptions for the data elements that the programmer must use to create the application functionality accurately minimizes costly development time.

Data Dictionary is a central repository for database Meta data. Based on the above facts, it is evident that the Data Dictionary helps to increase sharing, integration and reusability of data.

Normalization is the process of optimizing the storage and retrieval of information in a database. Normalization is a set of rules which help us to optimally design a database to reduce redundant data and unforeseen scalability issues. Mr. E.F.Codd, an employee of IBM, introduced the normalization process in 1971. To convert a database in a normal form, he proposed a serious of checkpoints for a database to follow. Initially, he introduced first three checkpoints called first, second and third normal form. These normal forms help to break a database into logical groups and tables that are more manageable. Later, Mr. Codd worked with Mr. Boyce and introduced a fourth and fifth normal form.

Let us assume that the following table is designed without following the normalization process.

Worker

Name	**Section**	**Required Skills**	**Section Manager**	**Section Strength**
David	Development	Java Visual Basic Oracle	Scott	30
Raj	Quality	WinRunner LoadRunner	Kim	15

In the above worker table, David has three skills and working in development section. The manager for development section is Scott. The above table design creates the following challenges:

- If a new employee wants to work for Quality, we need to create a third record that duplicates most of the information from the second row.
- If we want to add an attribute to reflect the level of ability for each skill, it is difficult to enter and maintain that data because each row has more than one skill.
- If Scott is replaced by a new hire, someone needs to edit all the records where the section manager is Scott.

The normalization process helps us to avoid the above challenges. The following sections explain objectives of each normal form and its implementation in the above example.

First Normal Form restricts each record's attribute to hold more than one value. In addition, to reduce a duplicate data, it requires moving the related information into separate tables and giving each table a primary key.

In the above example, to comply with the first normal form, the Required Skills field should be moved into the separate table called Skill as follows:

Worker (Primary key: Name)

Name	Section	Section Manager	Section Strength
David	Development	Scott	30
Raj	Quality	Kim	15

Skill (Primary key: Required Skills)

Required Skills	Section
Java	Development
Visual Basic	Development
Oracle	Development
WinRunner	Quality
LoadRunner	Quality

Second Normal Form requires that each non key attribute in a table depend on the entire primary key**.** In addition, the table must satisfy the first normal form.

In the Worker table, the Section Strength attribute does not depend on the primary key. To comply with the second normal form, the Section Strength attribute should be moved into the separate table called Section as follows:

Worker (Primary key: Name)

Name	Section	Section Manager
David	Development	Scott
Raj	Quality	Kim

Section (Primary key: Section)

Section	Section Strength
Development	30
Quality	15

Third Normal Form requires that each non key attribute in a table depend only on the primary key and no transitive dependencies in a table**.** In addition, the table must satisfy the second normal form.

In the Worker table, the Name attribute depends on the Section attribute and the Section Manager attribute depends on the Section Attribute. Since there is a transitive dependency between the Name attribute and the Section Manager attribute, the worker table violates the third normal form. To comply with the third normal form, the Section Manager attribute should be moved into the Section tables as follows:

Worker (Primary key: Name)

Name	**Section**
David	Development
Raj	Quality

Section (Primary key: Section)

Section	**Section Manager**	**Section Strength**
Development	Scott	30
Quality	Kim	15

After applying the first, second, and third normal forms to the Worker table, the result contains three tables as follows:

Worker (Primary key: Name)

Name	**Section**
David	Development
Raj	Quality

Section (Primary key: Section)

Section	**Section Manager**	**Section Strength**
Development	Scott	30
Quality	Kim	15

Skill (Primary key: Required Skills)

Required Skills	**Section**
Java	Development
Visual Basic	Development
Oracle	Development
WinRunner	Quality
LoadRunner	Quality

Fourth Normal Form restricts a record not to have two or more independent multi-valued facts about an entity. In addition, the table must satisfy the third normal form. **Fifth Normal Form** helps to decompose an entity without losing any potential information. In addition, the table must satisfy the fourth normal form.

Let's revise the initial challenges and how are they addressed by the normalization process:

- If a new employee wants to work for Quality, we need to create a new row in the Worker Table. The relation between the Worker and the Section produces the rest of information.
- If we want to add an attribute to reflect the level of ability for each skill, it is easy to create a separate table with the level of ability, and link it back to the Worker table.
- If Scott is replaced by a new hire, someone needs to edit only one row in the Section table.

After analyzing the above results, it is evident that the normalization process provides some great advantages. It reduces the redundant information which saves disk space and cost. Without building complex SQL statements, we can retrieve or store information. The normalization process reduces the complexity of a database design. Since the redundant information split into related groups, it is easy to understand and develop a logical and physical design of a database. In addition, the normalization process improves the security and performance. Grouping a related information helps to trouble shoot the security and performance issues without going through the entire database.