Finalizing Database Design

The next step in the database design process is to implement the logical database model into a physical database structure. First, the design team must translate the logical data model into a schema that the DBMS can interpret. In most cases, the schema will consist of an SQL data definition language (DDL) script. Because the logical and conceptual data models allow for design flexibility, quite a few database design elements can be modeled in such a way that they do not translate efficiently into a physical implementation. The following issues must be addressed at this point of the design process if they have not been resolved during previous steps.

If many-to-many relationships still exist within the database design, the following steps will resolve the issues:

1. Create a new table. It will be referred to as an association table.
2. Take the primary key from each table that participated in the M:M relationship, and use these keys to create a composite primary key in the association table.
3. Name the association table.
4. Create a relationship between the new association table and each table from the original M:M relationship. Each new relationship will be a one-to-many relationship, with the association table being the child in each.

Multivalued attributes are attributes that can contain more than one value. This signifies that an error was made within the normalization process. The normalized tables will have to be evaluated to resolve any issues.

Multipart attributes should be deconstructed into smaller parts if possible.

Derived attributes, or calculated attributes, should be removed, and the base value used for the calculation should remain within the model. The derived value can then be determined when information is retrieved by using the SQL SELECT statement.

By now, table names, column names, primary keys, relationships, and foreign-key relationships have been established. Other final tasks that need to be taken into consideration include the following:

- Designating specific data types for each column
- Deciding which attributes are required
  - If an attribute is required, then a NOT NULL constraint needs to be added.
- Implementing business rules, such as domain constraints and check constraints

Before the logical model is translated for a specific DBMS, the database project team must be aware of the following:

- A CHAR data type is a fixed-length object. If a column is defined as
CHAR(32), then the physical file will only store a total of 32 characters, regardless of what is stored in the column.

- A VARCHAR is a variable-length object. If a column is defined as a VARCHAR(32), then only the space needed for the string is stored in the file.
- A table with a foreign key must be created after the parent table.
- Physical implementations do not react well to space characters. When translating from a logical or conceptual model to the physical model, spaces should be changed to underscores or removed altogether.

Depending on the selected DBMS, the database administrator (DBA) might have the option to place individual tables on independent files, disks, or even controllers. When creating the tables, consideration must be given to the amount of system resources that two or more heavily used tables will consume.

The last database elements that need to be addressed when creating the table structures are known as indexes, which are created to improve data retrieval performance from SQL queries. An index is an object that is related to a database table but that is treated as a separate structure. If a column is used in the WHERE clause of an SQL statement, then it is an ideal candidate for an index. By default, any unique and primary key constraints will have unique indexes created automatically. Most foreign key columns should also have an index created against them.