Database Design & General Understanding

October 2006

Drew Nathanson
drew@technicalsynergy.com

What is a database?

A database is a collection of information stored in a computer in a systematic way, such that a computer program can consult it to answer questions. The software used to manage and query a database is known as a database management system (DBMS). Databases allow us to store information that can retrieve at a later date.

Kinds of Databases

There are several different kinds of databases:

- Rational
- CODASYL DBTG
- VSAM
- VTAM
- ISAM

Rational Database Names

- Oracle
- Microsoft
- DB/2
- Informix
- MySQL
- SyBase
- ProGres
- FreeBase
- And many more

Database Modeling Programs

There are several different database modeling programs that help you correctly determine how a database should be structured. The following is a list of those utilities:

- XCase
- ERwin
- Ferret
- UML
- ER Diagram Modeler

What they all have in common

Now we know that most of you are using at least one of the database mentioned above or something similar to handle all your data needs. Although most of them behave somewhat differently (and perform there processes differently), they all have many things in common. They are:

- They all support Structured Query Language (SQL) to the ANSI standards committee.

- They all support the ability to have primary and foreign keys besides indexes.
- They all support the relational model:
    - One to One
    - One to Many
    - Many to Many
    - Many to One
- They all provide the ability to create logical tables.

Remember, most databases provide the ability to program within the database itself and it is incumbent on your to learn those process to make your applications more efficient.

Designing a Database

Usually the people who design databases are called Database Designers (DB Designers). DB Designers will examine all the ancillary factors into account when designing a database. For example, they would look at Disk Space required, performance, hardware, volume, etc. Refer to the Data Warehousing to more details. Data Modelers take the project requirements and create the database according to best practices for the given system. This usually includes table structure, indexes, etc. Their job also requires an understanding how the data will be used and by whom. For example, a data modeler would need to know if the application that is using the data is a 3-tier application as apposed to an enterprise wide application. The data may the same but the retrieve process is somewhat different.

For many of us in the ColdFusion arena, our job as developer also includes being DBMS administration and database modeler. This is the purpose of this document is help clarify what are the best practices for database design.

Back to Basics

The basic design of a relational database is that *like* information is grouped together to form a single unit of information (called records) that is stored in a Table. Each record would be uniquely identified and would be able to be retrieved via the SQL process.

A collection of *like* tables is called a schema. A sub-set of the total number of tables is a called a sub-schema. Usually these refer to areas where access restrictions are required.

Remember, that databases were designed to handle large amount of data and allow for the fastest way to retrieve it. The use of primary keys is an efficient way to perform this. The other ways to improve performance is to index the most used fields. Indexes are used by the DBMS system and allow the system to take advantage of the power database to retrieve information based on indexes within the system.

We will be discussing this further in the Normalization section.

We now have a basic understanding of how we want to organize our data. The challenge to the database is how to make the different areas relational. As the design process progresses the relationship between tables becomes much clear. Remember, the design of a database is a re-iterative process. Each cycle gets you close to the goals.

Using Primary Keys

One of the major advantages to relational databases is the ability to the use Primary Key in order to identify an individual record. The primary key is field (or fields) and must be by definition unique.

Foreign Keys & Referential Integrity

Foreign keys are another advantage of relational databases that allows you to ensure that primary keys from other tables stored in a record (to allow relational joining) are indeed correct.

Referential integrity is a database concept that ensures that relationships between tables remain consistent. When one table has a foreign key to another table, the concept of referential integrity states that you may not add a record to the table that contains the foreign key unless there is a corresponding record in the linked table. It also includes the techniques known as cascading update and cascading delete, which ensure that changes made to the linked table are reflected in the primary table.

Indexes

Relational databases allow for the ability to index specific fields within a table to help make the querying of data much easier. Indexes are physical entities and therefore affect performance.

Physical & Logical Objects

There is much confusion over the definition of what is a physical record verses what is a logical record. For ease of understanding, the following is a summary of the types:

- A table and associated keys and indexes are physical records within the database.
- All views and pieces of data retrieved are logical records and are created when requested (and not before).

Downside of Relational Database

Well, relational databases certainly provide a vast way for us to retrieve and examine data, so what are the drawbacks?

The following is a list of the downsides of Relational Databases:

- Knowing the data – One of the main drawbacks to relational databases is knowing what data to keep.

- User Expectation – The end user of the data needs to know what there expectations are going to be. These expectations could be about any number of areas – such as what is returned, performance, readability, search ability, etc.

- Performance – A poorly designed database will take longer to return the needed requirements.

- Hardware – Depending on the application, the hardware that the database used could also be a large factor is providing a response.

Data Warehouse

A data warehouse is a collection of data gathered and organized so that it can easily by analyzed, extracted, synthesized, and otherwise be used for the purposes of further understanding the data. It may be contrasted with data that is gathered to meet immediate business objectives. Such immediate data will ultimately become a part of the warehouse.

The generally accepted range for warehousing to begin is around a million records. However, there is more to warehousing then just the number of records. In order to create an effective warehouse work correctly, usually the data needs to be re-organized to make the querying of the data efficient.

Here are some guidelines what looking at warehouses:

- The limited use of indexes
- De-normalization of the data to 1st Normal Form
- Aging the data
- Physical hardware

The other large part of data warehousing that is critical is the use of SQL. The more complex the SQL statements are, the longer the query will take.

Finally, user expectation is one very important part of what are acceptable performance constraints.

Normalization

Normalization is the process of organizing your data and breaking it into smaller tables that are easier to manage. The primary reason for normalizing a database is to prevent data redundancy. The goal of database normalization is to ensure that every non-key column in every table is directly dependent on the key, with the benefit in the form of reduced redundancies, fewer anomalies and improved efficiencies.

Database design theory includes standards and guidelines that help efficiently design the tables and create a normalized database. The following is the list of the guidelines:

- First Normal Form (1NF):
    - Break each field down to the smallest meaningful value
    - Remove repeating groups of data
    - Create a separate table for each set of related data

- Second Normal Form (2NF):
    - Create new tables for data that applies to more than one record in a table
    - Add a related field (foreign key) to the table

- Third Normal Form (3NF):
    - All columns in the table depend on the primary key.

Now all of the sounds wonder but what does it really mean to ColdFusion developers who have 4 weeks to create a web site?

Refer to the different ER diagrams for how we will normalize a database.

Recommendations

Since most of us are using databases in order to perform the functions of our websites, it would in your best interest to learn how to utilize the more advanced features of the database.

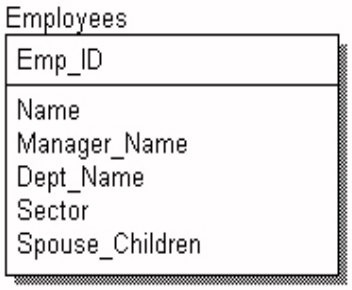
The more advanced features include areas like:

- Functions
- Stored Procedures
- Triggers

These features can be incorporated into your application and will make processing the data much easier.

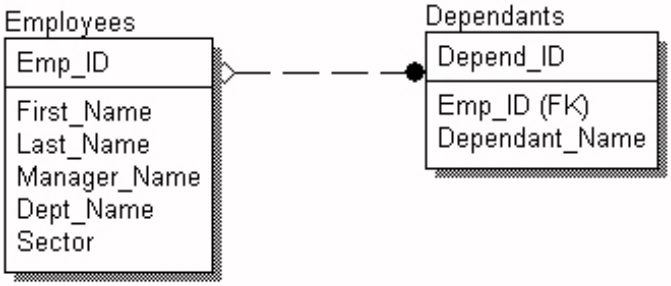
Normalization of a Relational Database

Database Table as Defined



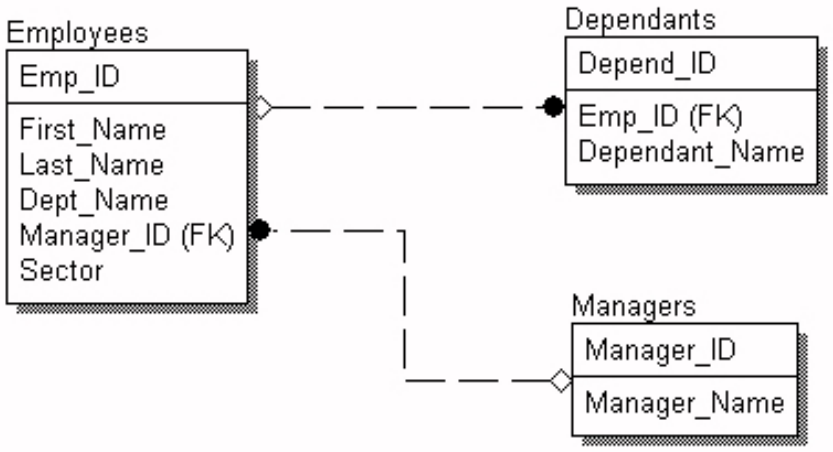
Normalization of a Relational Database

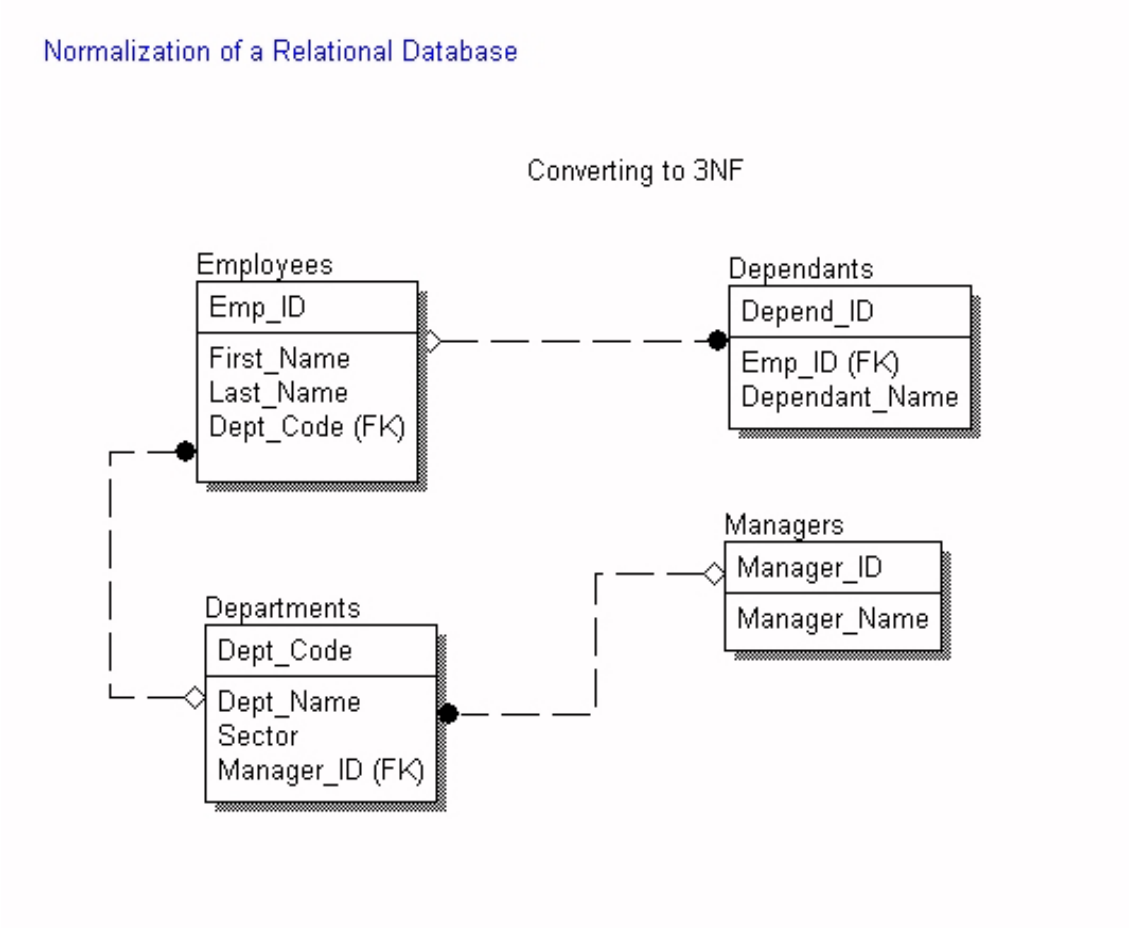
Converting to 1NF



Normalization of a Relational Database

Converting to 2NF




Normalization of a Relational Database
Converting to 3NF
Employees
Emp_ID
First_Name
Last_Name
Dept_Code (FK)
Dependants
Depend_ID
Emp_ID (FK)
Dependant_Name
Managers
Manager_ID
Manager_Name
Departments
Dept_Code
Dept_Name
Sector
Manager_ID (FK)