Code and Coffee - Database Basics

Karl Benedict, Jon Wheeler. UNM Libraries, Research Data Services

February 9, 2018

Why Use a Database to Organize Your Data

- Consistent structure - defined by you
- Enforced data types
- Can scale from single tables to sophisticated relational data models
- Can be a personal file-based or shared server-based solution, depending on your needs
- Standard language for interacting with your data
- "Virtual tables" can be created on the fly based on database queries
- Data can be accessed by many analysis tools

Individual Topics

1. What is a table?
2. What is a database?
3. Different database solutions
    a. File-based (personal)
    b. Server-based (personal or team)
4. Client applications (interact with both file- and server-based solutions)
5. Structured Query Language (SQL) - *mostly-standard* (Demo)

What is a table?

Superficially, a table is a set of values defined by rows and columns. Columns can be variously thought of as domains, variables, characteristics, etc. that apply to each item in a set of entities. In this case each row (AKA *record* or *tuple*) represents an instance of or object within that entity set.

Donuts make a good example. Let's imagine that we've decided to classify an assorted dozen donuts. Three obvious characteristics stand out - the dough, the glaze, and the filling. Every donut in our dozen can be described according to these characteristics and represented with a unique row in a table:

donutID	dough	glaze	filling
1	cake	maple	none
2	yeast	sugar	none
3	yeast	maple	boston creme
4	yeast	chocolate	none
5	cruller	vanilla	none
6	yeast	chocolate	boston creme

Easy! BUT - table design can be one of the more interesting and challenging aspects of data definition and database development. Donuts make an easy example because a box of donuts is not necessarily a complex system. It's a box. With donuts. Compare that with Amazon.com or other online retail system within which a large number of complex entities interact in complex ways, or with a census of animal populations at the Bosque del Apache. Would you record all the animals in a single table? Use one table for birds and one mammals, or even create a separate table for each species?!

Without getting into the weeds of normal forms or normalization, it's important when designing a database to give thought to:

- The sets or types of entities described
- Unique characteristics of each set
- Overlapping or shared characteristics among sets
- Cardinality between sets

What is a database?

In many respects a database is a collection of **one or more tables** that contain **data related to a specific problem** that enables **management of the contained data** and the **ad-hoc retrieval of data/information** from the database. This management and information retrieval is through integrated functions that support the *definition* and *modification* of the database structure, and the *creation*, *reading* (query), *updating*, and *deletion* (CRUD) of records in the database. The key characteristics of a database that must be determined include:

1. What tables do we need to create that will store the data of interest?
2. What are the useful relationships between tables that we need to consider when we think about how to break our dataset into logical pieces? What types of relationships do we need?
3. What types of data will go into the individual columns of each table in the database? Numeric? Textual? Dates/times? Locations? Logical?
4. What information do we need to get out of the database? Specific reports? Visualizations?

5. Are there any ways we can increase the performance of our database by preprocessing the information in the database to speed the retrieval of specific data? (i.e. define indices)

To illustrate some key database concepts we will now walk through a more complex database design that illustrates some key aspects of the thinking for and implementation of a database solution for a research problem.

The problem

We would like to gather and organize the data needed to track trends in donut type selection for sale through time. To answer this question we need data relating to donut shops, their menus, and the donuts they sell.

With this problem in hand we can start to identify the data we will need to address this problem and sketch out the database design we might use to organize the data once we are able to obtain it. The following figure illustrates a proposed database design that we can walk through to illustrate the key database characteristics outlined above.

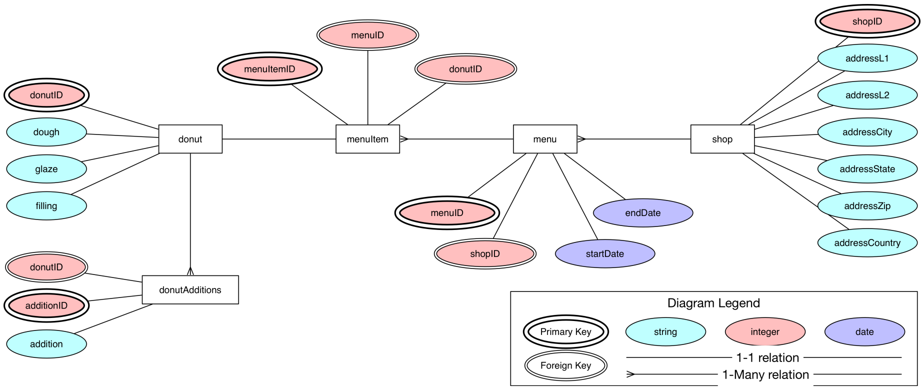

Figure 1: Donut database design (link to full-resolution image)

Aside - What about Excel?

Managing your data in Excel is sort of like living in parts of the mythical Wild West:

1. You are free to do pretty much anything you want
2. There's a lot of manual labor
3. If you're not careful you could shoot your foot off
4. You or a member of your family could get bit by a snake

Figure 2: "The Outlaws" - photo by Magnus - https://flic.kr/p/e5HGcE. CC BY-ND 2.0

Different database solutions (a sample)

One way to think about different database systems is to distinguish between **file-based databases** that are essentially based upon a file (or files) that you directly interact with on your **local computer**, and **server-based databases** that *may* run on your local computer or on a remote system and typically **expose their capabilities through network services**. File-based databases are most commonly focused on a *single-user* model while server-based databases are typically designed to support multiple concurrent users.

File-based (personal)

- Filemaker Pro
- SQLite (SpatiaLite)
- Microsoft Access
- dBase (included primarily for historical reasons)
- LibreOffice Base

Server-based (personal or team)

- Oracle
- MySQL
- Microsoft SQL Server
- PostgreSQL
- IBM DB2

Check here for a more comprehensive list of relational database options.

Client applications (interact with both file- and server-based solutions)

While many relational database systems implement the core *Structured Query Language* for interacting with a database and its contents. (SQL will be demo'ed next), some databases either extend the language with their own proprietary extensions, or use their own language all together to perform database management and query tasks.

A wide variety of client applications exist that can communicate either directly (through the implementation of capabilities to interact with a file- or server-based database without using an intermediate operating system tool like ODBC [see below]), or indirectly with databases to allow for the use of specialized interfaces or tools for database management or data analysis.

In an effort to lower the barriers to connecting between applications and databases the *Open Database Connectivity* (ODBC) standard was developed to define a standard method for applications to interact with an *ODBC driver* that is set up on the local computer to enable communication with a specific database. If an application implements support for ODBC it can use an available ODBC driver to connect to any supported database.

Commonly used analytic tools that can connect to database servers include:

- Spreadsheet applications (such as Microsoft Excel)
- Geographic Information Systems (such as ArcGIS or QGIS)
- Statistical tools (such as R, SPSS, SAS)
- More general analysis tools and languages (such as Matlab, Python)
- Web programming and publication languages (such as PHP)