**Database Architecture and Design**

Nate Bachmeier  
October 8th, 2013  
DBM 380: Database Concepts  
University of Phoenix  
Instructor: Sandra La Vallee

# Database Systems

Since the 1970s, databases have been gaining popularity in both the academic and businesses circles (Coronel, Morris, & Rob, 2013). Some common use cases for these data stores is to provide backend systems to: web sites, CRM systems, maintain application state, or local caches (Roussopoulos, 1991). But what is a “database?”

“Databases, are specialized structures that allow computer-based systems to store, manage, and retrieve data very quickly (Coronel, Morris, & Rob, 2013).” These specialized structures come in numerous forms and architectures, each with different goals in mind.

How users interact with the database depends on the system design. For instance relational databases typically use Structured Query Language (SQL); which “defines data structure and […] provides functional capabilities for creating, accessing, maintaining, controlling and protecting data (International Standards, 1992).” Another common language is UnQL, which is has characteristics of SQL and JavaScript. It enables easy users to easily query and update unstructured data in systems such as NoSQL (Williams, 2012).

# Database Architectures

There are many ways to categorize and group databases, among the many dimensions include: supported number of connections, degree of schema enforcement, and logical storage model.

## Connection Dimension

When it comes to number of connections databases can range from single user to hundreds of thousands. For example, Microsoft SQL Server Compact is a database system which is intended for “building standalone and occasionally connected applications for mobile devices, desktops and Web clients (Microsoft DDC, 2013).” With SQL Compact applications can host the database directly in the process. At the other end of the spectrum is Oracle 12c Enterprise Server, which is hosted in the cloud and supports an unlimited number of connections.

## Schema Enforcement Dimension

The next dimension enforcement of schema; comes in one of two flavors: Structured, and Semi Structured. Of these structured schema systems are the most common for business database systems. By leveraging schema products such as MySql, SqlLite, and SQL Server; can perform type safety checks and better ensure data integrity.

Some databases only need to store documents or other blobs which will only have loose definitions (Buneman, 1997). For these semi structured scenarios optimizations can be made to give users, of products such as Neo4j or Mongo DB; more flexibility with data retrieval.

## Logical Storage Dimension

The third dimension presented is dependent on the logical storage model of the database. Two of the more interesting models are relational and graph based.

The origin of relational databases can be traced back to the 1970 paper by Codd; his model contained abstractions of the physical and logical storage model. In addition it defined that all data be stored in tables (relations) as rows (sets) and column (tuples). Then using mathematical principals it is possible to quickly and efficiently apply predicate calculus to consistently retrieve specific information (Codd, 1970).

An alternative system to relations based on graph theory, was creatively named Graph Databases. These first appeared during the 1980s but faded out during the 1990s. Then during the 2000s systems such as social networks created a need for this unique storage model (Angles & Gutierrez, 2008). In such a system it is possible to traverse entities in terms of arbitrary relationships instead of simply following foreign keys (Barceló Baeza, 2013).

# Uses in the Workplace

At work we primarily use Microsoft SQL Server as our database of choice. Some of the many scenarios include:

* Maintaining Application State
* Reporting on Long Running Progress
* Keeping Data Consistent as our team spans three continents

# Conclusion

Databases come in several shapes and sizes to quickly snap into any business or application scenario. These variations can be grouped based on the dimensions of simultaneous connections, schema enforcement, and logical storage. Given the vast number of options it is important to understand the differences of each system so the best tool for the job can be selected.

# References

Angles, R., & Gutierrez, C. (2008). Survey of graph database models. *ACM Computing Surveys (CSUR) Volume 40 Issue 1*.

Barceló Baeza, P. (2013). Querying graph databases. *PODS '13 Proceedings of the 32nd symposium on Principles of database systems*, 175-188.

Buneman, P. (1997). Semistructured data. *PODS 97 Proceedings of the sixteenth ACM SIGACT-SIGMD-SIGART symposium on Principles of database systems*, 117-121.

Codd, E. (1970). A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM*, 377-387.

Coronel, C., Morris, S., & Rob, P. (2013). *Database Systems: Design, Implementation, and Management, Tenth Edition.* Cengage Learning.

International Standards. (1992, July). *Information Technology - Database Language SQL.* Retrieved from Carnegie Mellon University: http://www.contrib.andrew.cmu.edu/~shadow/sql/sql1992.txt

Microsoft DDC. (2013). *Sql Server Compact*. Retrieved from Data Developer Center: http://msdn.microsoft.com/en-us/data/ff687142.aspx

Roussopoulos, N. (1991). Modern client-server DBMS architectures. *ACM SIGMOD Volume 20 Issue 3*, 52-61.

Williams, P. (2012, March 29). *UnQL: A Standardized Query Language for NoSQL Databases*. Retrieved from DataVersity: http://www.dataversity.net/unql-a-standardized-query-language-for-nosql-databases/