Database Types

Overview

Traditionally many enterprise systems stored their data in relational databases. In recent years, the drive for greater scalability, high throughput and large datasets has led to the adoption of alternatives. The following table lists types of database and example implementations of each.

	Description	Examples
Relational	Data organised into table and rows	SQLServer, PostgreSQL, Sybase
Document		MongoDB
Graph		Neo4j, InfiniteGraph
Columnar		Cassandra, HBase

The alternatives to relational databases were typically introduced to provide the following benefits.

Why Non-SQL?

- Greater scalability than relational databases
- Support exceptionally large data sets
- Support extremely high throughput

Relational Versus NoSQL

While NoSQL is an overly broad category encompassing many different types of data store we can still generalize to the situations where Relational stores are better than NoSQL and vice versa.

Table 1 Properties

	Relational	**NoSQL**
Structure	Hard to add columns after data	Columns can be added after data and not all objects need values for all columns
Scalability	Only easily scaled vertically by adding memory and compute. Horizontal scaling hard	Easily supports horizontal scaling. Can be deployed to the cloud and most implementations support horizontal scaling out of the box
Consistency	ACID transactions	No ACID transactions to improve scalability and performance

Comparing the properties of Relational and NoSQL databases gives us an indication of when we might use each. Relational databases are good when the following are true.

- We need ACID transactional consistency.
- Our data schema is well defined and fairly static.

In contrast we would look to NoSQL databases in situations where the following conditions hold.

- We need horizontal scalability using the cloud.
- Storing large amounts of non-homogenous objects.