NoSQL

- "Not only SQL"
- Designed for modern big data environment
  - Volume- terabytes/day
  - Velocity- events/sec
  - Variety- not necessarily simple types
- How does NoSQL address these challenges?
  - Scalability- distribute across nodes
  - Availability
    - Multiple replicated nodes with failover
    - Tradeoff: expensive to ensure consistency between replicas
    - Evenutal consistency
  - Sharding- distribute across nodes
  - Key lookup
  - No schemas- using semi-structured, self-describing data (JSON and XML)
  - Less powerful query langauges- simple Create-Read-Update-Delete (CRUD) operatons
    - No joins
    - Single object access
- Major Categories of No SQL
  - Document based
  - Key value stores
  - Columnar: store columns in files, not tables
  - Graph-base
  - Combinations of the above

Eventual Consistency

- NoSQL opts for eventual consistency, meaning propagation of updates to all nodes
- Impacts 'I' in ACID (isolation)
- Performance advantage

CAP: Consistency, Availability, Partition Tolerance

- Consistency: across replicas
- Availability: every request to database is answered
- Partition tolerance: ability to keep functioning when netowrk is partitioned due to fault
- CAP theorm: not possible to guarantee all three CAP properties at the same time in a distributed system with data replication