Dr Greg Wadley

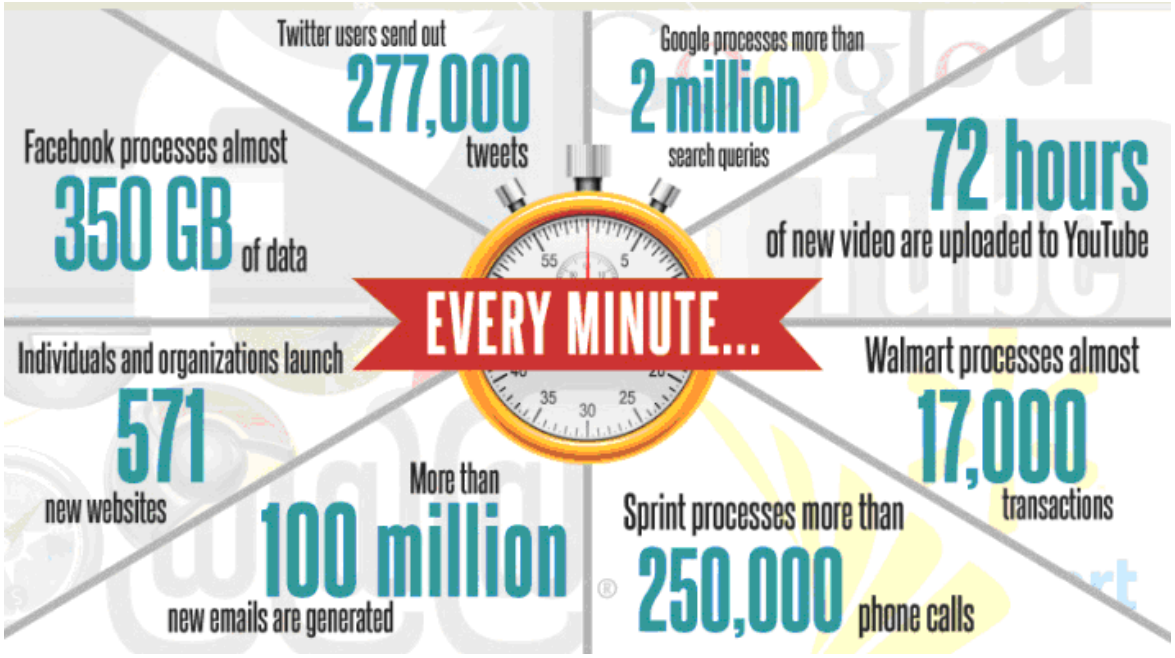
INFO90002 Database Systems & Information Modelling

Week 12

Database Trends

Today's Session…

- The Relational status-quo
- Challenges:
  - Cloud storage
  - Big Data
  - Object-orientation
- The NoSQL response

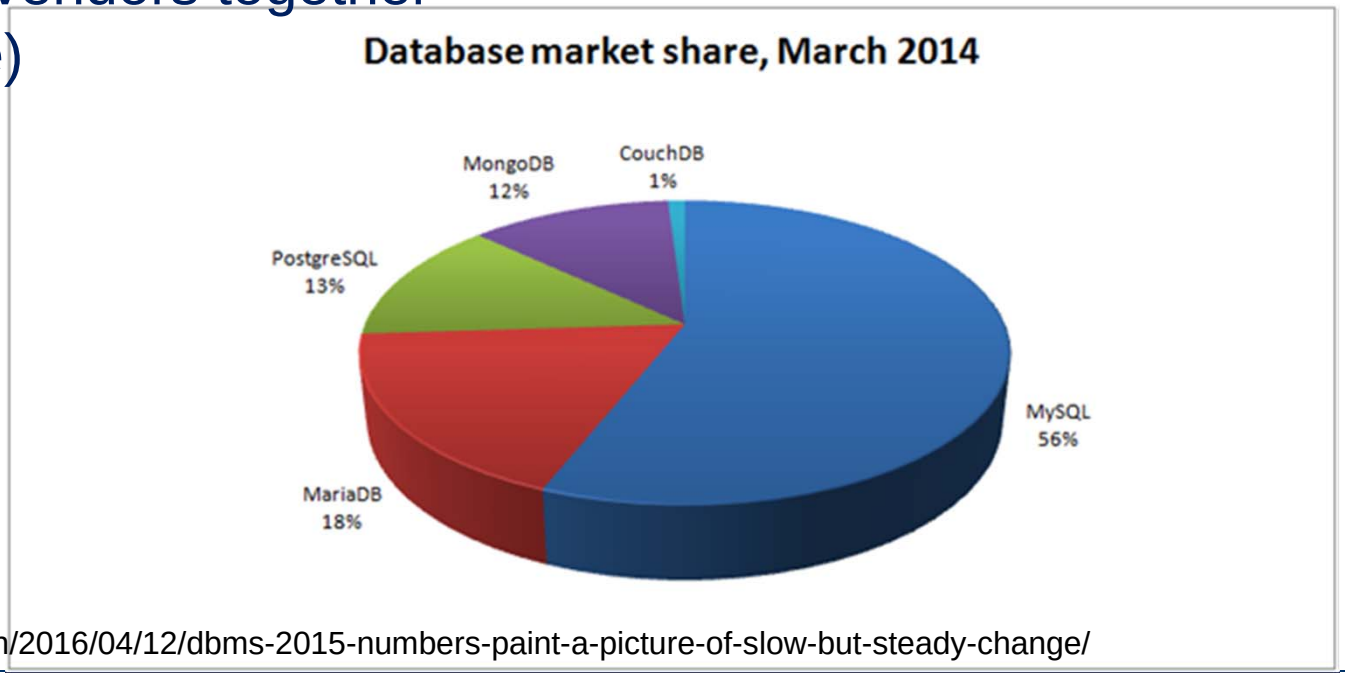


The Relational status-quo
- major vendors
- current offerings

Leading RDBMS vendors

- Top *commercial* RDBMS (Gartner, 2015)
  - 41.6% Oracle
  - 19.4% Microsoft Sql Server
  - 16.5% IBM DB2
  - (top 3 own over ¾ of market)
  - (top 5 NoSQL vendors together are at 8th place)

Open source DBMS



http://blogs.gartner.com/merv-adrian/2016/04/12/dbms-2015-numbers-paint-a-picture-of-slow-but-steady-change/

RDBMS popularity

- **Number of mentions of the system on websites**, measured as number of results in search engines queries. At the moment, we use Google and Bing for this measurement. In order to count only relevant results, we are searching for <system name> together with the term database, e.g. "Oracle" and "database".
- **General interest in the system.** For this measurement, we use the frequency of searches in Google Trends.
- **Frequency of technical discussions about the system.** We use the number of related questions and the number of interested users on the well-known IT-related Q&A sites Stack Overflow and DBA Stack Exchange.
- **Number of job offers, in which the system is mentioned.** We use the number of offers on the leading job search engines Indeed and Simply Hired.
- **Number of profiles in professional networks, in which the system is mentioned.** We use the internationally most popular professional network LinkedIn.
- **Relevance in social networks.** We count the number of Twitter tweets, in which the system is mentioned.

Gartner's "Magic Quadrant"

"By 2017, all leading operational DBMSs will offer multiple data models, relational and NoSQL, in a single DBMS platform.

Through 2018, a wave of consolidation will affect the operational DBMS market's smaller vendors, through mergers, acquisitions and business failures.

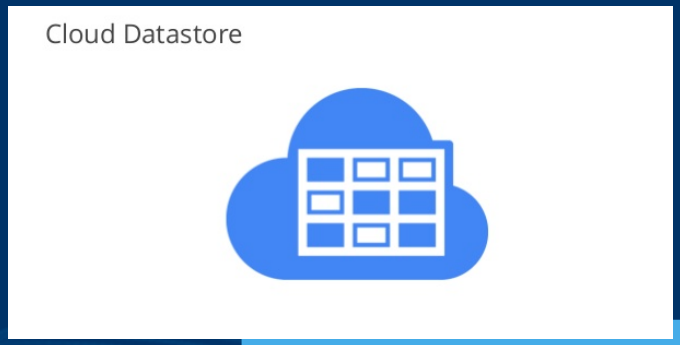
By 2017, the "NoSQL" label will cease to distinguish DBMSs, which will result in it falling out of use."

(Gartner, 2015)

https://www.gartner.com/doc/reprints?id=1-2PMFPEN&ct=151013&st=sb

Current major relational offerings

Challenge: Cloud storage



Database as a service (DBaaS)

- Similar to other cloud services
  - DBMS and data are in the cloud
  - your application connects as required
  - pay per usage, quantity data, input/output
  - DBMS is administered by provider
    - reduces need for in-house DBA
    - may be managed via web console
- Can be presented as either
  - Database as a service (DBaaS)
    - relational database
    - non-relational (NoSQL) database
  - Virtual machine with a database installed
- Advantages
  - Simplifies setting up and especially scaling up your database

Potential cloud db issues

- Data security
  - provider may not have fully integrated security structure
  - may need to resort to encryption
- Legal frameworks
  - need to (continue to) conform to laws governing use of data
    - Can the data be moved out of a political jurisdiction?
    - How must the data be secured?
- Large movements of data between your site and cloud
  - during initial setup
  - during ongoing integration with local data

sourced from InfoWorld (2012)
http://www.infoworld.com/d/cloud-computing/the-unpleasant-truths-about-database-service-208450

Example cloud database vendors

- Information from www.mitre.org/work/tech_papers/2012/11_4727/cloud_database_service_dbaas.pdf
- SQL
  - Amazon Relational Database Service
  - Microsoft SQL Azure
  - Oracle Cloud
  - Rackspace
- NoSQL
  - Google Datastore
  - Amazon SimpleDB

Comparison of cloud db

	Amazon RDS (MySQL)	Microsoft SQL Azure	Google Datastore	Amazon SimpleDB
Type	**RDBMS**	**RDBMS**	**NoSQL**	**NoSQL**
Maximum amount of data that can be stored	1 terabyte per database[2]	50 gigabytes per database[3]	Not published for entire database, but 1 MB limit on a subset of data (called an entity). Limit to the number of indexes.	10 gigabytes per database domain (roughly equivalent to an RDBMS table)[4]
Ease of software portability with similar, locally hosted capability	High. MySQL instantiation in cloud is very similar to the local instantiated version.	High. Most SQL Server features are available in SQL Azure.	Medium/Low. Requires Java Data Objects or Datastore-specific interface and use of App Engine.	Medium. Requires SimpleDB-specific interface.
Transaction capabilities	Yes	Yes	Yes	Yes
Configurability and ability to tune database	High. MySQL instantiation in cloud.	Medium. Can create indexes and stored procedures, but no control over memory allocation or similar resources.	Low	Low
Database acces-sible as "stand-alone" offering	Yes	Yes	No. Requires Google App Engine application layer.	Yes
FISMA Certified	No	No	No	No
Can designate where data is stored (e.g., region or data center)	Yes	Yes	No	Yes
Replication	Yes	Yes	Yes	Yes

Pricing & Performance

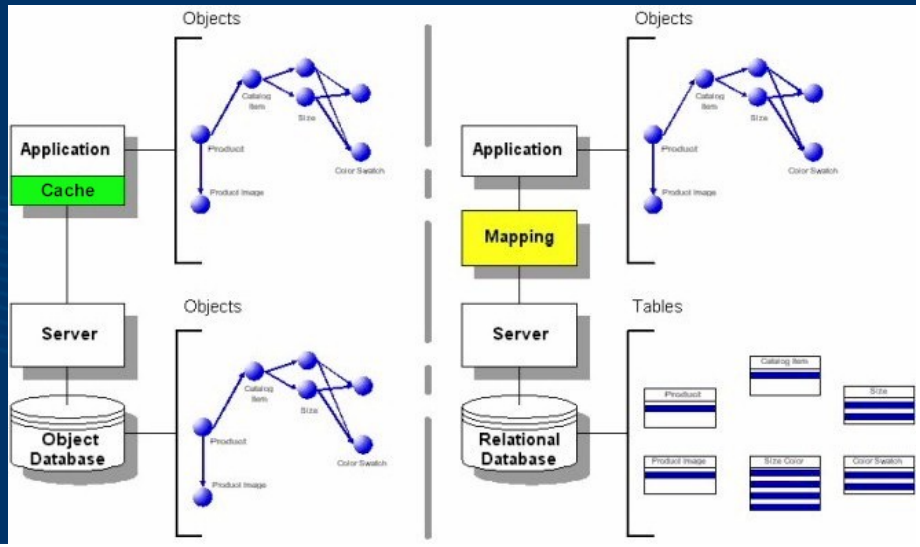
	Amazon RDS (MySQL)	Microsoft SQL Azure	Google Datastore	Amazon SimpleDB
Example Pricing for Processing (Refer to Sections 2-5 for details)*	Ranges from $0.11 per RDS hour for smallest instance to $2.60 per hour for largest instance	Ranges from $9.99 per database with up to 1 GB of storage to $499.95 per database with up to 50 GB of storage per month	$0.10 per App Engine CPU hour (required for accessing Datastore)	$0.14 per SimpleDB unit hour
Example "On-demand" Pricing for Data Transfers (Refer to Sections 2-5 for details)*	Inbound $0.10 per GB and outbound ranges from $0.15 per GB to $0.08 per GB, depending on volume	Inbound $0.10 per GB Outbound $0.15 per GB	Inbound $0.10 per GB Outbound $0.12 per GB	Inbound $0.10 per GB and outbound from $0.15 to $0.08 per GB, depending on volume.
Example Monthly Pricing for data storage (Refer to Sections 2-5 for details)*	$0.10 per GB plus $0.10 per 1 million I/O requests	Included in processing pricing	$0.15 per GB	$0.25 per GB

Single Client	Average writes per second	Average reads per second
Amazon MySQL RDS*	2,567	2,551
Microsoft SQL Azure	406	410
Google Datastore**	288	200
Amazon SimpleDB	208	63

5 Clients	Average writes per second	Average reads per second
Amazon MySQL RDS*	7576	7905
Microsoft SQL Azure	1737	1893
Google Datastore**	N/A	N/A
Amazon SimpleDB	689	281

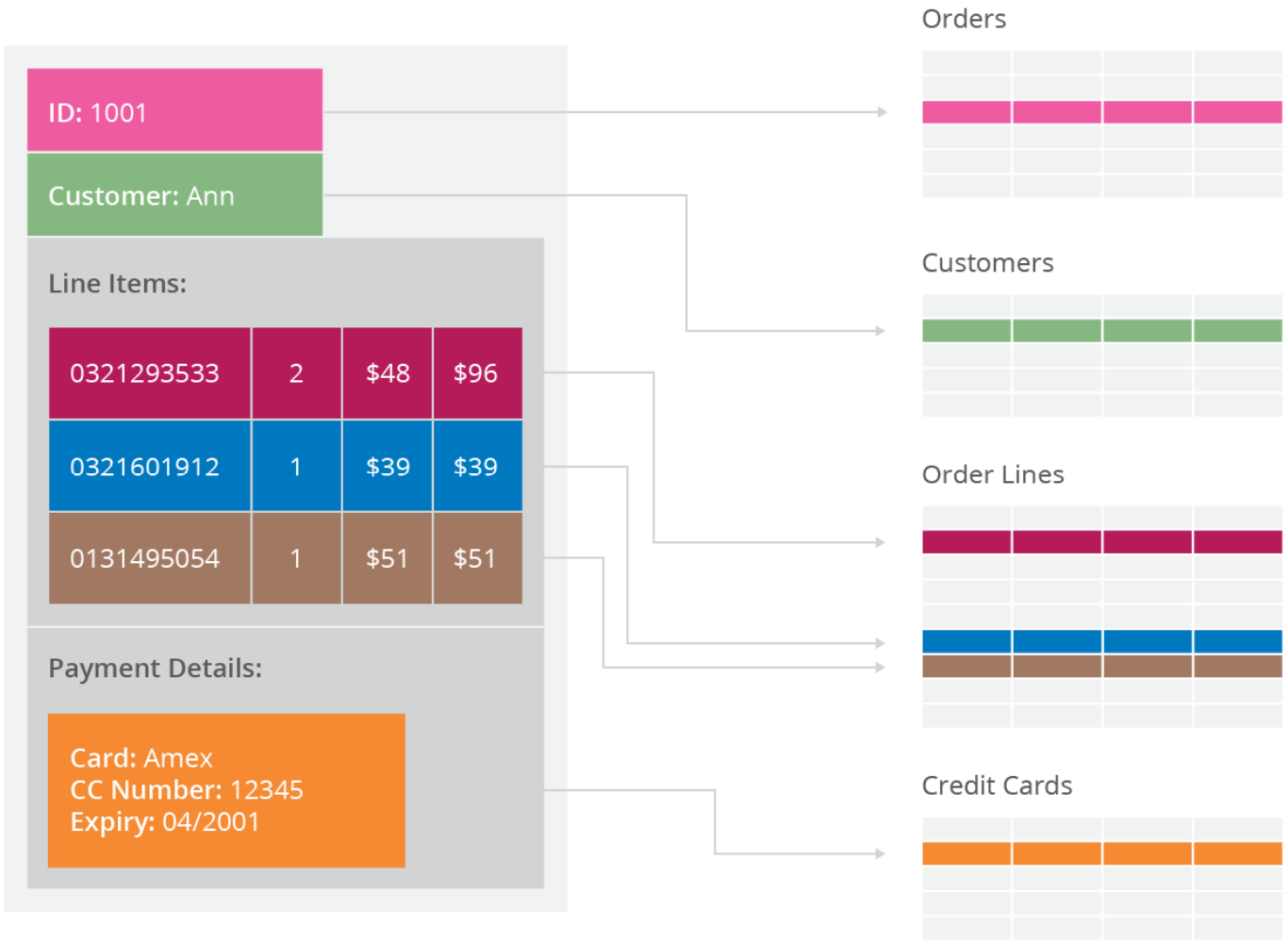
THE UNIVERSITY OF MELBOURNE

Challenge: Object-orientation



Object-Relational impedance mismatch

One aggregate in memory must be stored across many tables



Relational vs OO mismatch

- History
  - relational DB rose to prominence in the 1980s
  - object-oriented programming (OOP) rose later in the 1990s
- Mismatch
  - OO allows more complex data structures
  - complex objects must be ‘normalized’ into relational tables
- How can an OO program persist data in a database?
  - relational database (simple, powerful, but not OO)
  - object-relational databases (complex data types, SQL 1999)
  - object-oriented databases (did not become popular)
  - object-relational mapping software (currently popular)
  - NoSQL (becoming popular in some niches)

Object-oriented DBMS

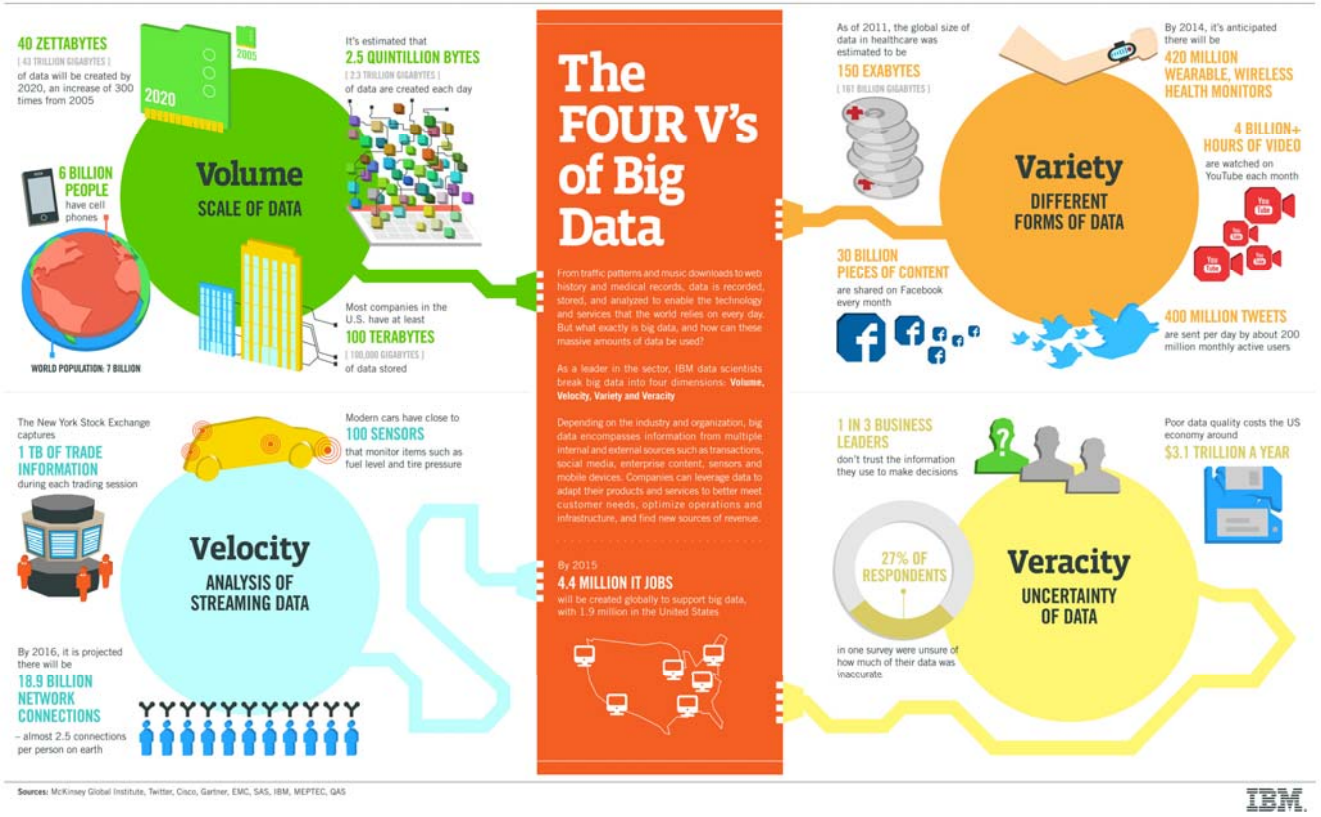
- Direct storage of object-oriented data
- Offers benefits of OO and Relational
  - directly persists objects
  - offers performance and security of relational DB
- Products include:
  - Db4o, InterSystems, McObject, Objectivity, Progress, Versant, ObjectDB 10gen, Gemstone, VelocityDB …
- But did not become popular in the marketplace
  - often tied to particular OO programming languages
  - lack of compatibility between different OODMBSs
  - lack of standard ad hoc query language
  - RDBMS are used to integrate applications
  - (but, re-emerging in the form of NoSQL ‘document databases’)

Challenge: Big Data

"Big Data is a huge trend and is very real. Behind the label, however, what Big Data really means is new database and data processing technologies, leading to a giant inflection point for database technology, the biggest in 25 years. Part of Big Data is, of course, the sheer volume of data (i.e., the "big" part). But part of the trend toward Big Data is that innovative database technology today is demonstrably "better." (10gen CEO, creator of MongoDB)

Extremely large data sets

- Difficult to process high volume and velocity of data using traditional database management and processing tools
- Challenges include capture, storage, search, sharing, analysis, and visualization.



Why Big Data presents a challenge

- Relational databases emerged 40 years ago when data was generated within organizations, mostly by employees, at a human scale, and often manually entered.
- This scenario still occurs! Here, relational is still best.
- But technical innovations mean data is now generated by: social media, global website clickstreams, machinery, sensors, scientific instruments…
- "Internet of Things" … 30 billion wireless devices by 2030 (http://www.dbta.com/Editorial/Trends-and-Applications/Powering-the-Internet-of-Things-with-Real-Time-Hadoop-103469.aspx)
- Exabytes … 1,000,000 terabytes 1,000,000,000 gigabytes
- Volume and velocity of data is too high for SQL databases

Solution: distributed database

- Large operations like Google and Amazon were not able to store their data in relational databases
- How to scale up?
- “Bigger servers” gives diminishing returns
- Therefore spread data across many small servers
- But relational not good with distributed data
- So these companies devised new databases that are designed from the ground up to run across multiple servers
- Google: *Bigtable*
- Amazon: *Dynamo*
- This inspired the NoSQL movement

Big Data – problems

- In its infancy
  - tools for using and analysing big data, as well as standards, are still being developed
- Need the right talent and technology and structure of workflows to optimize the use of big data
- Requires expensive professionals ("data scientists")

  "There will be a shortage of talent necessary for organizations to take advantage of big data. By 2018, the United States alone could face a shortage of 140,000 to 190,000 people with deep analytical skills as well as 1.5 million managers and analysts with the know-how to use the analysis of big data to make effective decisions" (McKinsey Global Institute)
- https://mbs.edu/programs/master-of-business-analytics
- http://www.msi.unimelb.edu.au/study/graduate/master-of-data-science/

material in this section is drawn from
http://martinfowler.com/books/nosql.html,
including talk at GOTO conference 2012

The NoSQL response

"The database market is back in play after a 30-year old freeze in which Oracle dominated the high end, and Microsoft the mid market. Then along came open source, the cloud, NoSQL, in memory and everything changed….The idea that everything is relational? Those days are gone."

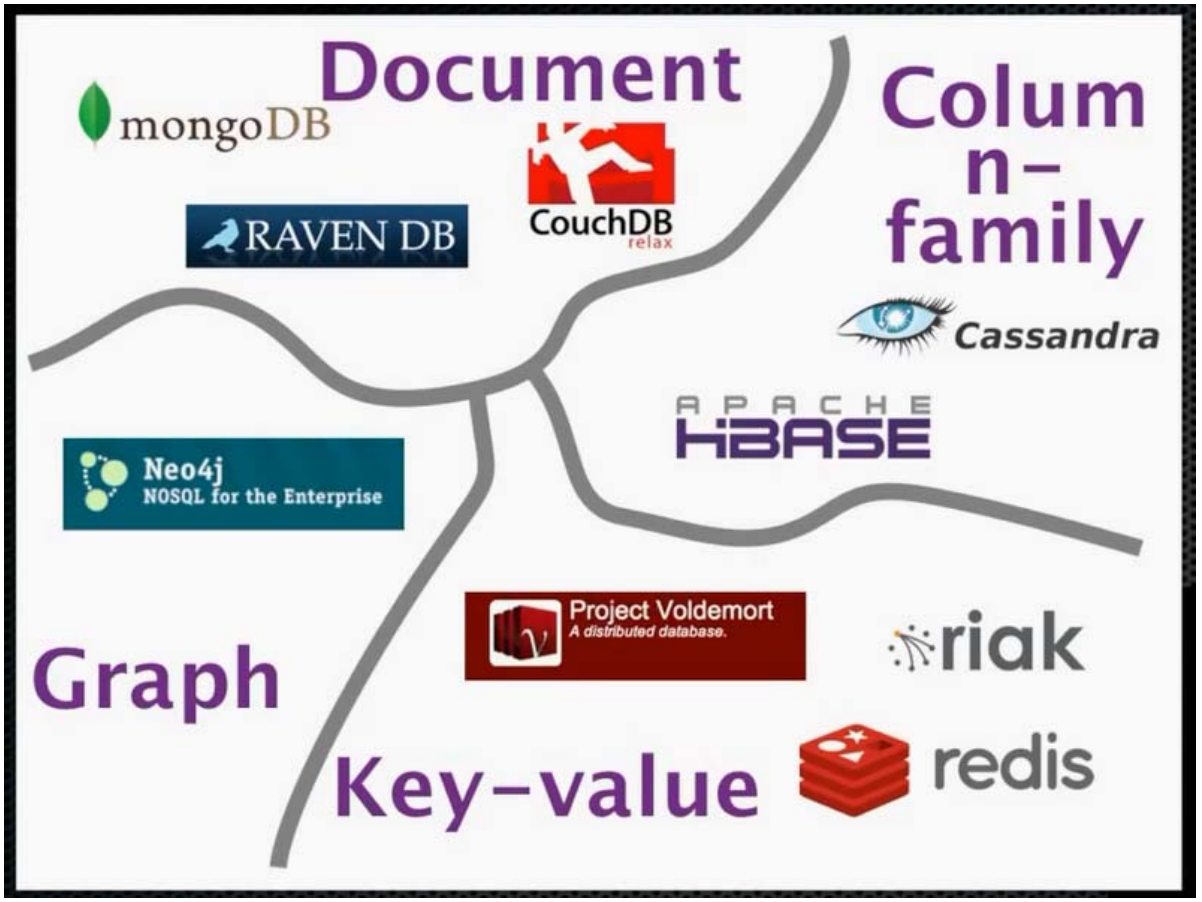
Redmonk analyst James Governor

NoSQL databases

- Features
  - doesn’t use relational model or SQL language
  - runs well on distributed servers
  - most are open-source
  - built for the modern web
  - schema-less (though there may be an "implicit schema")
  - 'eventually consistent'
- Goals
  - to improve programmer productivity (OR mismatch)
  - to handle larger data volumes and throughput (big data)

from *NoSQL Databases: An Overview*
by Pramod Sadalage, Thoughtworks (2014)

Types of NoSQL database



(extra NoSQL material has been moved to separate lecture)

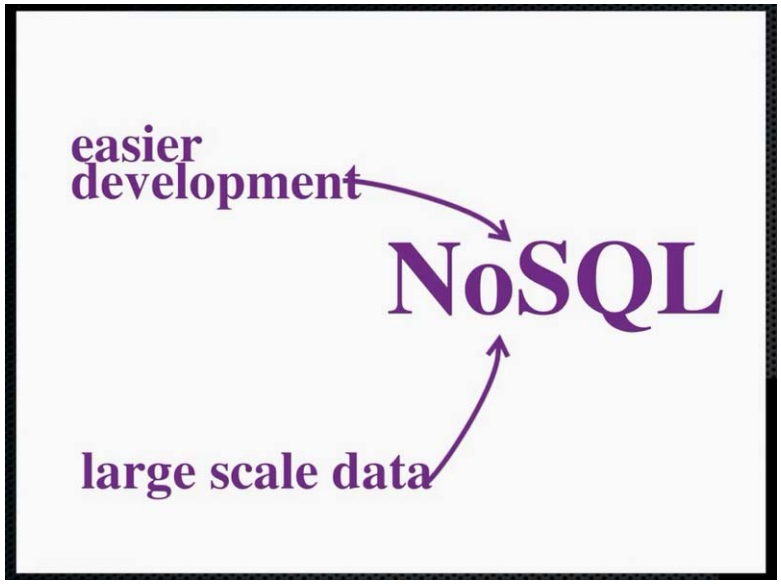
Some notable NoSQL users

- Google – BigTable
  - search, gmail, maps, youtube
- Facebook – Cassandra, Tao, Giraph
  - messaging, social graph
- Amazon – SimpleDB, DynamoDB
  - large scale e-commerce and analytics, cloud db
- Instagram - Cassandra
  - social media newsfeed
- LinkedIn – CouchDB, MongoDB
  - monitoring and analysis of operational data
- The Guardian - MongoDB
  - newspaper articles, user identity
- FourSquare- MongoDB
  - venues and user checkins

Do you need to know NoSQL?

Q. Do only big web companies like Google, Amazon and Facebook need NoSQL?

A. In fact, any organization is likely to have to start dealing with large amounts of data (due to web, mobile, sensors etc), while some are adopting NoSQL to avoid object-relational mismatch (making programming easier).



But NoSQL tends to be for niche situations, and RDBMS will probably continue to serve most applications.