FIT1043 Introduction to Data Science

Week 10

Ian K T Tan

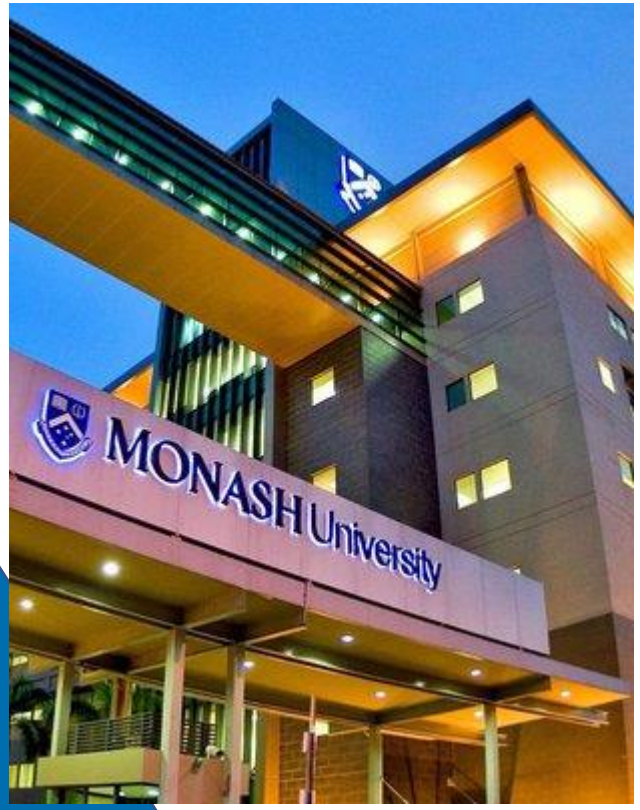
School of Information Technology
Monash University Malaysia

With materials from Wray Buntine, Mahsa Salehi

Week 9 Coverage

Characterising Big Data

Shell Scripts

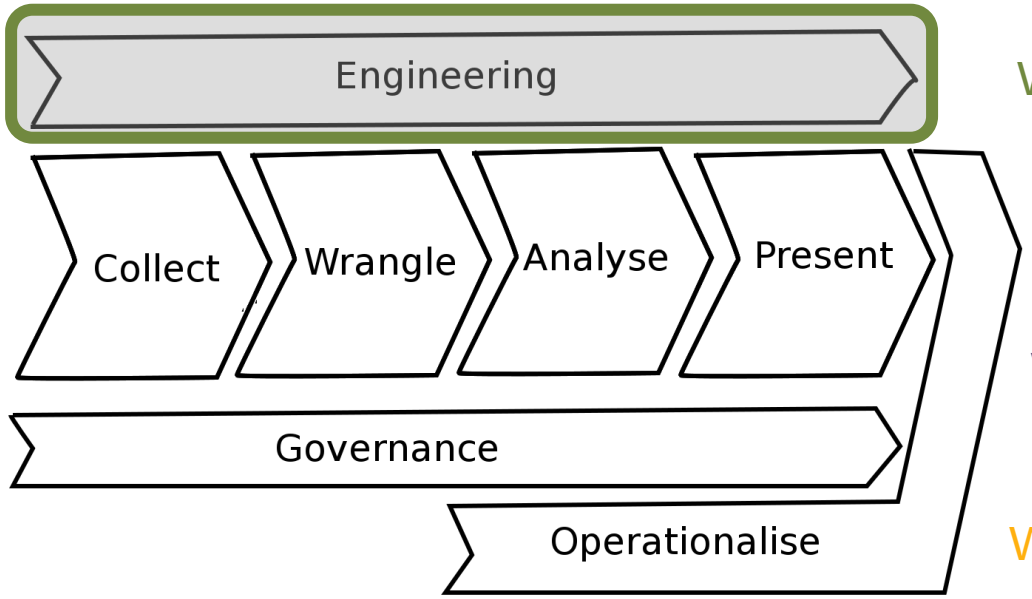


MONASH University

Week	Activities	Assignments
1	Overview of data science	
2	Introduction to Python for data science	
3	Data visualisation and descriptive statistics	
4	Data sources and data wrangling	
5	Data analysis theory	Assignment 1
6	Regression analysis	
7	Classification and clustering	
8	Introduction to R for data science	Assignment 2
9	Characterising data and "big" data	
10	Big data processing	
11	Issues in data management	Assignment 3
12	Industry guest lecture (tentative)	

Week 1

Overview of data science



Weeks 9-10

Week 4

Week 3

Weeks 5-7

Week 11

Weeks 2 & 8

Tools for data science

Big Data Processing

Week 10 Outline

Different databases

- Storing and accessing data

Introduction to distributed processing

- Map-reduce
- Hadoop
- Spark

Learning Outcomes

Week 10

By the end of this week you should be able to:

- Characterize different database types
- Differentiate between SQL and NoSQL databases
- Define what distributed processing is
- Analyse the Map-Reduce framework
- Differentiate between Hadoop and Spark
- *Apply R/shell commands to read/manipulate big data files*

Storing Data
Databases

MONASH University

Business Context

Businesses function in a continuously changing environment:

- Fixed formats as per RDBMS not suitable
- Usage varies, requires complex analytical queries

Need to reach insights faster and act on them in real time

- Stream processing

Databases

Storing and Accessing Data

Relational DataBase Management Systems (RDBMS)

- Typically SQL based

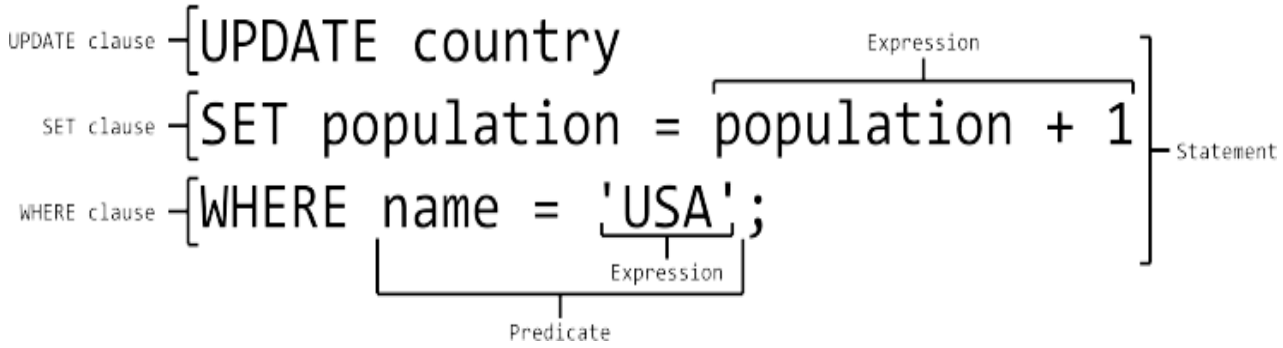
NoSQL

- Typically JSON storage format

Graph DataBase

Databases: RDBMS

SQL Structured Query Language



It is like a large scale set of Excel spreadsheets with better indexing and retrieval capabilities

Transaction oriented with support for correctness, distribution, and most importantly ACID (Atomicity, Consistency, Isolation, Durability)

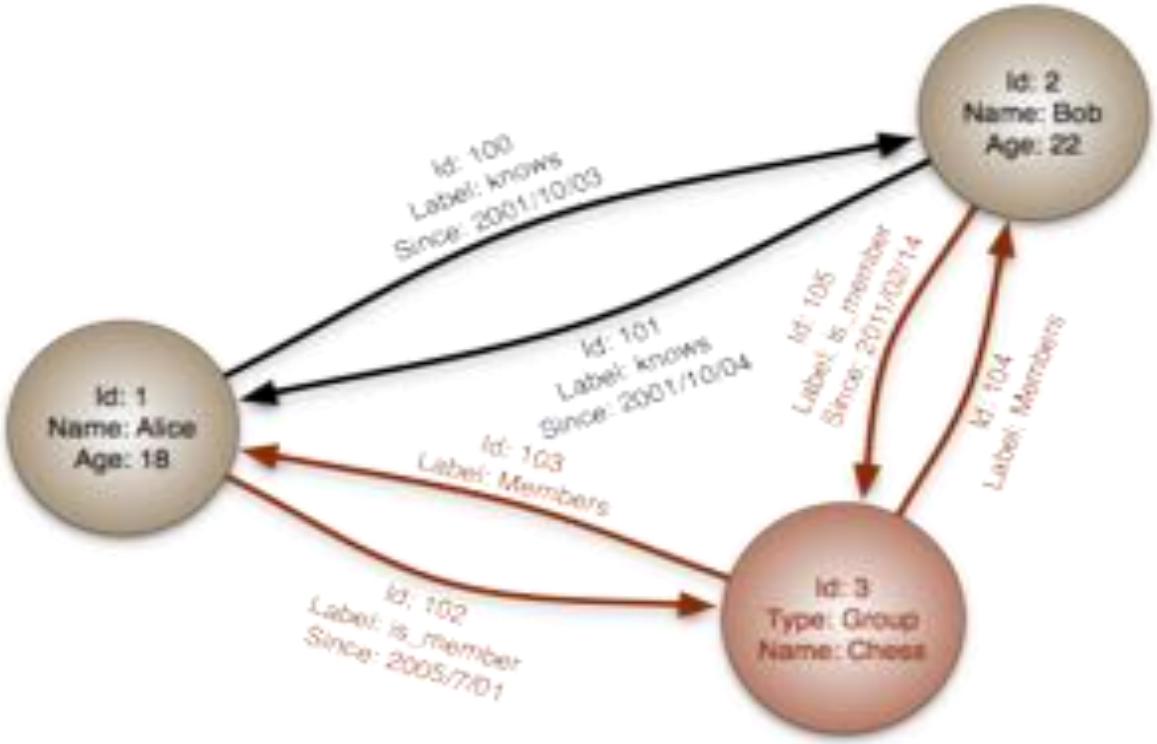
JSON Example

- No fixed format
- Semi-structured, key-value pairs, hierarchical
- “friendly” alternative to XML
- Self-documenting structure

```
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 25,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "office",
      "number": "646 555-4567"
    }
  ],
  "children": [],
  "spouse": null
}
```

Databases: Graph

- Stores graph, commonly as triples or subject, verb and object
- Commonly used to store Linked Open Data



NoSQL

From https://en.wikipedia.org/wiki/NoSQL

A NoSQL (originally referring to "non SQL" or "non relational") database provides a mechanism for storage and retrieval of data that is modeled in means *other* than the tabular relations used in relational databases. Such databases have existed since the late 1960s, but the name "NoSQL" was only coined in the early 21st century, triggered by the needs of Web 2.0 companies. NoSQL databases are increasingly used in big data and real-time web applications. NoSQL systems are also sometimes called "***Not only SQL***" to emphasize that they may support SQL-like query languages, or sit alongside SQL databases in polyglot persistent architectures.

Database Concepts

Background

in-database analytics:	the analytics is done within the DB (e.g. R in SQL Server)
in-memory database:	the DB content resides memory (e.g. GCP MemoryStore)
cache:	data stored in-memory (e.g. ElastiCache for Redis)
key-value:	value accessible by key, e.g., hash table (e.g Oracle NoSQL)
document based:	storing, retrieving and managing document-oriented information (e.g. MongoDB)

Database Concepts

Background

information silo: an insular information system incapable of reciprocal operation with other, related information systems

For example:

- If two big banks merge, then initially their RDBMSs will be siloed
- In a big insurance company, auto and home insurance customer RDBMSs may be siloed

SQL and Beyond SQL Databases (NoSQL)

Background

Many NoSQL and SQL DBs offer:

- Large scale, distributed processing
- Robustness
- General query languages
- Some notion of consistency, e.g. “eventually” as nodes spread updates

SQL and Beyond SQL Databases (NoSQL)

Suitability

- Use **SQL** database when:
  - data is **structured** and unchanging

- Use NoSQL database when:
  - Storing large volume of data with little to no structure
  - Data changes rapidly

Volume, Velocity, Variety

- NoSQL databases offer a rich variety beyond traditional relational databases.

SQL and Beyond SQL Databases (NoSQL)

https://www.youtube.com/watch?v=EE8ZTQxa0AM

MongoDB Video, but watch until 2:50 only

0:15 – 0:45, introduction to how structured data is store (like in Excel)

0:45 – 1:10, data contains many wasted space (sparsity of data)

1:10 – 1:40, build sheets/tables and connect (normalization)

1:40 – 2:25, explains the issue with this (although RDBMS is highly optimized for this)

2:25 – 2:50, how it is stored in a Document Based DB, such as MongoDB

Beyond SQL Databases (NoSQL)

Type	Notes
RDBMS	SQL
Object DB	Data stored in Objects, natural integration to OO languages
document based DB	JSON like (e.g. MongoDB)
key-val cache	In-Memory
key-val store	Not In-Memory, but highly optimized
tabular key-val	Relational-like, “wide column” store
graph DB	RDF, SPARQL

Summary: Databases Overview

Linkage vs Structural Complexity

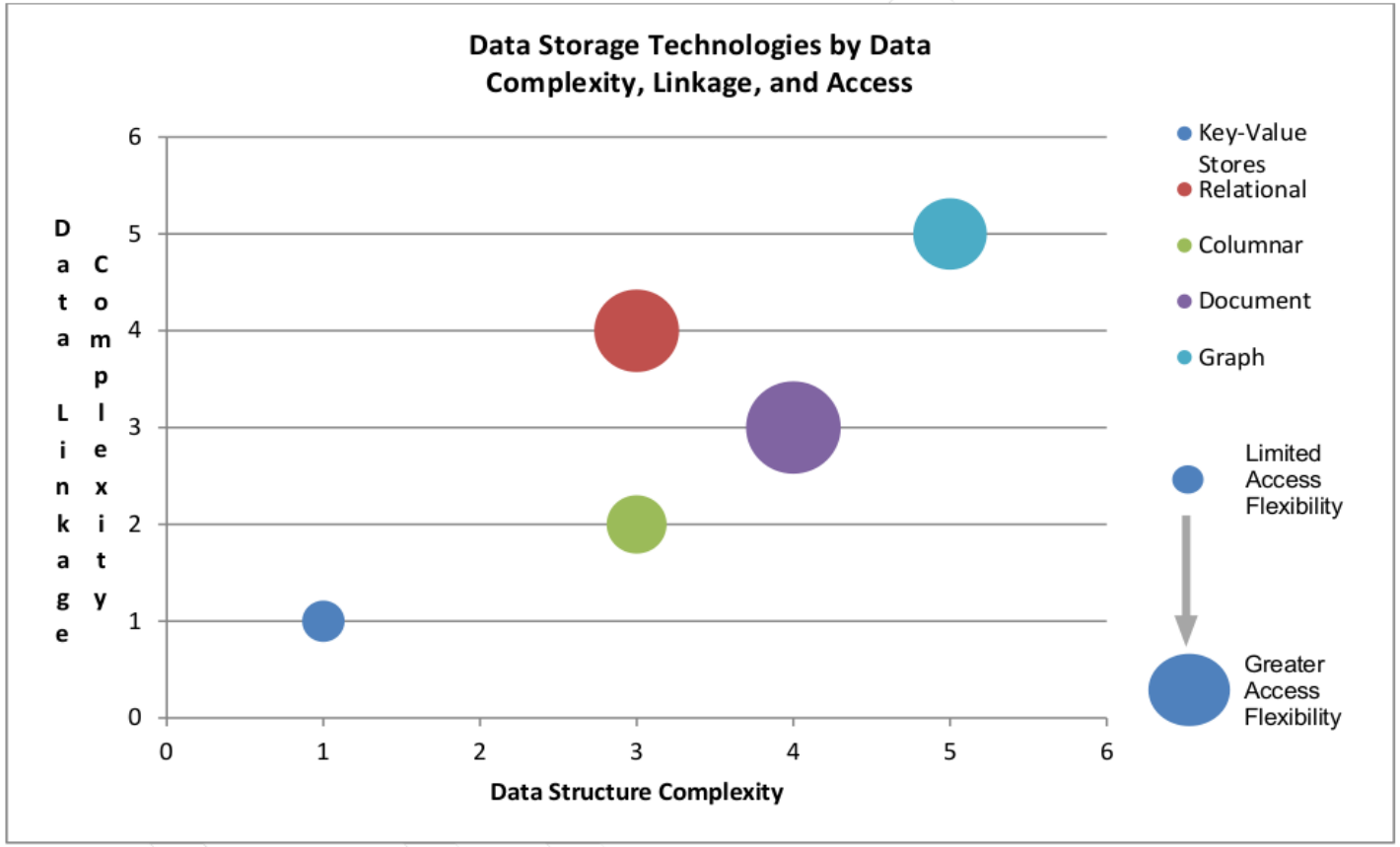


Figure 4: Data Storage Technologies