Griffith UNIVERSITY

1814ict/2814ict/7003ict:
Data Management/
Database Design

# Topic 6.1: Big Data, NO SQL

(Chapters 14)

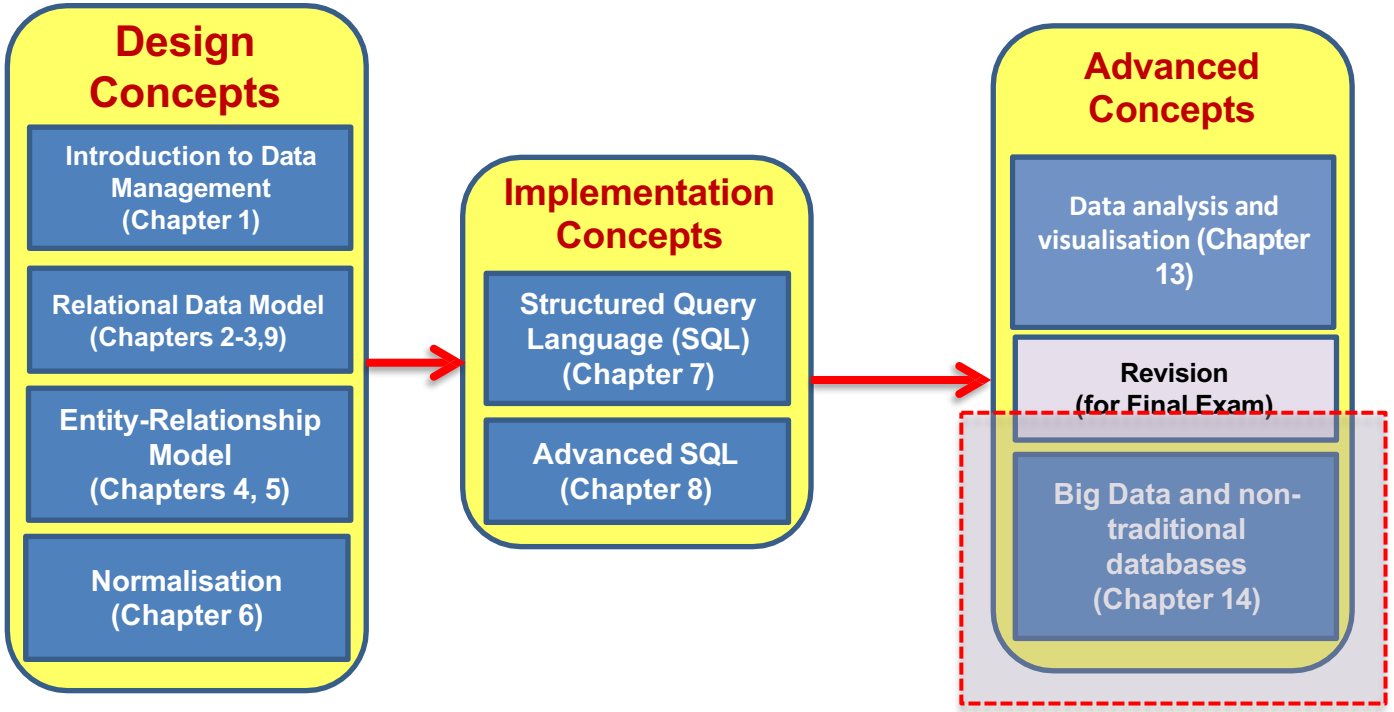
**Course convenor: AProf. Henry Nguyen**

School of Information and Communication Technology

**Course developed by**: Dr Mohammad Awrangjeb; AProf John Wang; Dr Zhe Wang

# Course bigger picture

- Chapter references are to textbook *Database Systems: Design, Implementation, & Management - By Carlos Coronel and Steven Morris*



# Learning Outcomes

At the end of this lecture students will be able to know:

- What is Big Data & why it is important
- What is NO SQL Database

# Content

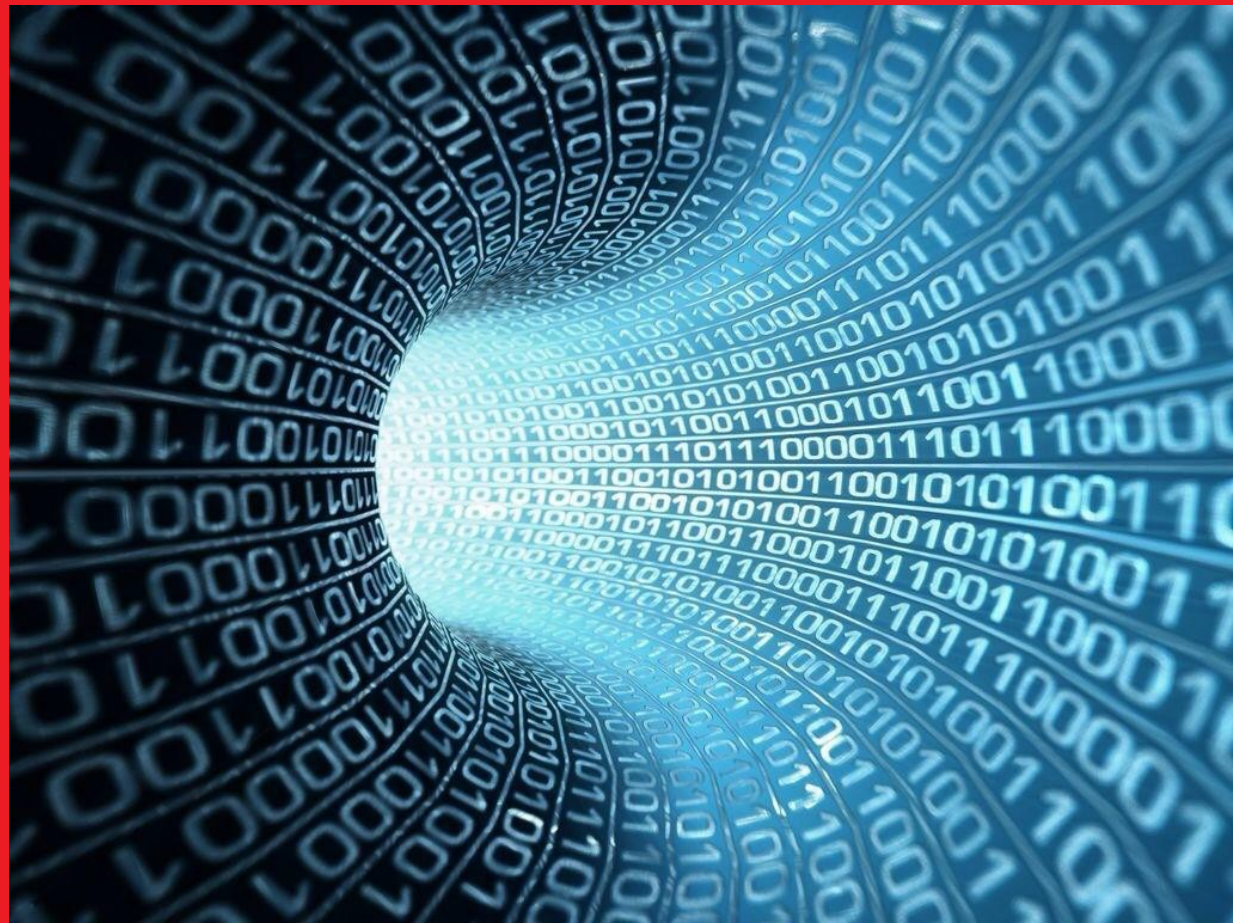
- What is Big Data
- Limitations of Relational Database

**Outcomes 1**

- Characteristics of NO SQL Databases
- Examples of NO SQL databases

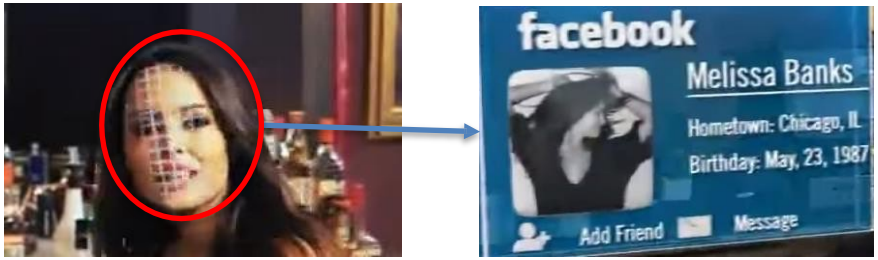
**Outcomes 2**

# BIG Data



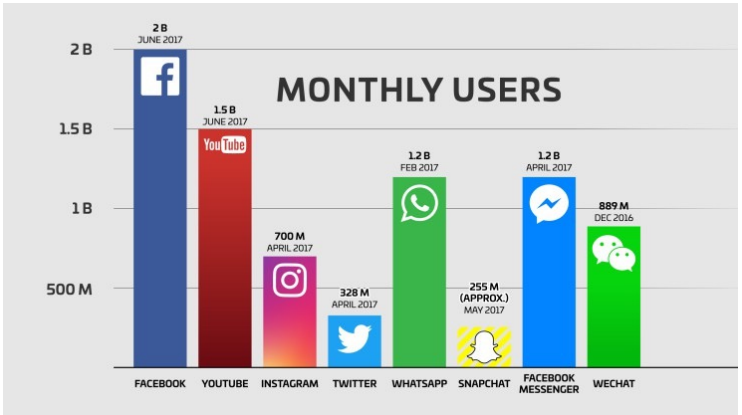
# How much data to process?

- Face recognition in AR video (Topic 1.1)
  - Without even the person notices!
- Face recognition to a huge database?
- Facebook!
  - How Big?



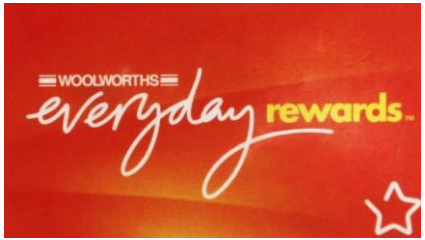

2016 Jan 27


2017 Jun 27



# Big Data

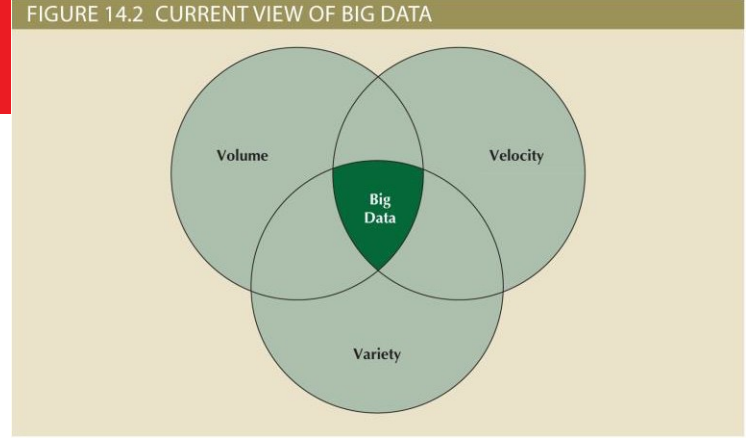
- What is Big Data?
  - Vast amounts of generally unstructured data
  - Distributed customer information collected from
    - Transactional histories
    - Customer feedback & surveys
    - Social media applications
    - Mobile device activities, and
    - Software logs
  - **Why important**:
    - Processed using special software to find new insights about customer behaviour
    - Predict future from current!
      - Flybuys, everyday rewards!
      - Toddler products ➔ School products!

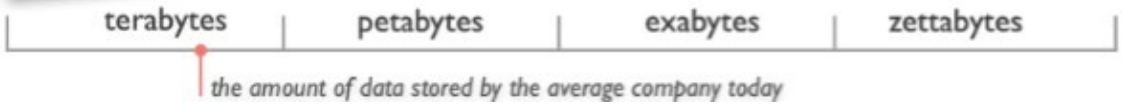



# What is BIG Data

## Based on 3Vs

- **Volume**: Quantity of data to be stored
  - E.g., Walmart handles >1 million customer transactions every hour, importing > 2.5 petabytes of data into DB, 167 times of the information contained in all the books in the US Library of Congress




**US congress library:**

- Founded in 1800 with 6,487 books
- Now >16M books & > 120M collections

- To address big volume:
  - **Scaling up** is keeping the same number of systems but migrating each one to a larger system
  - **Scaling out** means when the workload exceeds server capacity, it is spread out across a number of servers

**Multiples of bytes** v·t·e

| Decimal | | | Binary | | |
|---|---|---|---|---|---|
| Value | Metric | | Value | IEC | JEDEC |
| 1000 | kB | kilobyte | 1024 | KiB kibibyte | KB kilobyte |
| $1000^2$ | MB | megabyte | $1024^2$ | MiB mebibyte | MB megabyte |
| $1000^3$ | GB | gigabyte | $1024^3$ | GiB gibibyte | GB gigabyte |
| $1000^4$ | TB | terabyte | $1024^4$ | TiB tebibyte | – |
| $1000^5$ | PB | petabyte | $1024^5$ | PiB pebibyte | – |
| $1000^6$ | EB | exabyte | $1024^6$ | EiB exbibyte | – |
| $1000^7$ | ZB | **zettabyte** | $1024^7$ | ZiB zebibyte | – |
| $1000^8$ | YB | yottabyte | $1024^8$ | YiB yobibyte | – |

Orders of magnitude of data

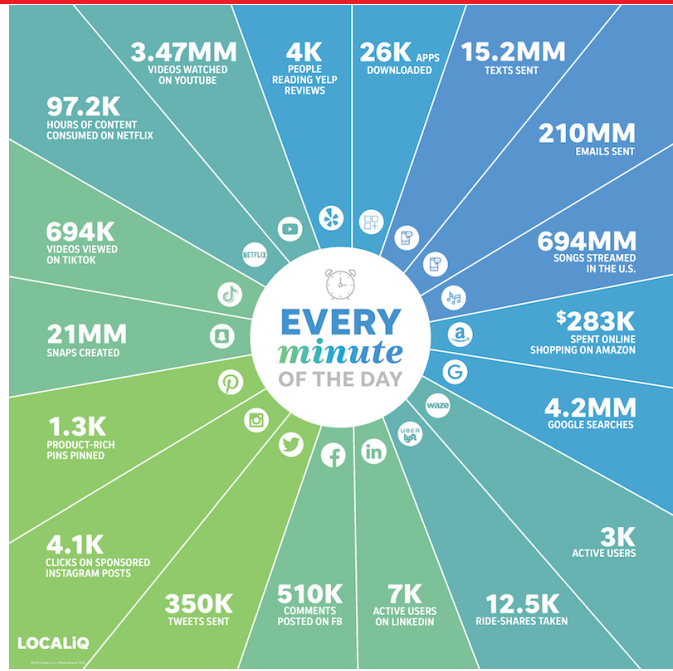
# What is BIG Data

## Based on 3Vs

- **Velocity:** Speed at which data is entered into system and must be processed

  Let's visit what happen in real-time at online!
  - http://www.smartinsights.com/internet-marketing-statistics/happens-online-60-seconds/

- New processing challenges:
  - **Stream processing** focuses on input processing and requires analysis of data stream as it enters the system
  - **Feedback loop processing** refers to the analysis of data to produce actionable results




**Every 60 seconds in 2021**

# What is BIG Data

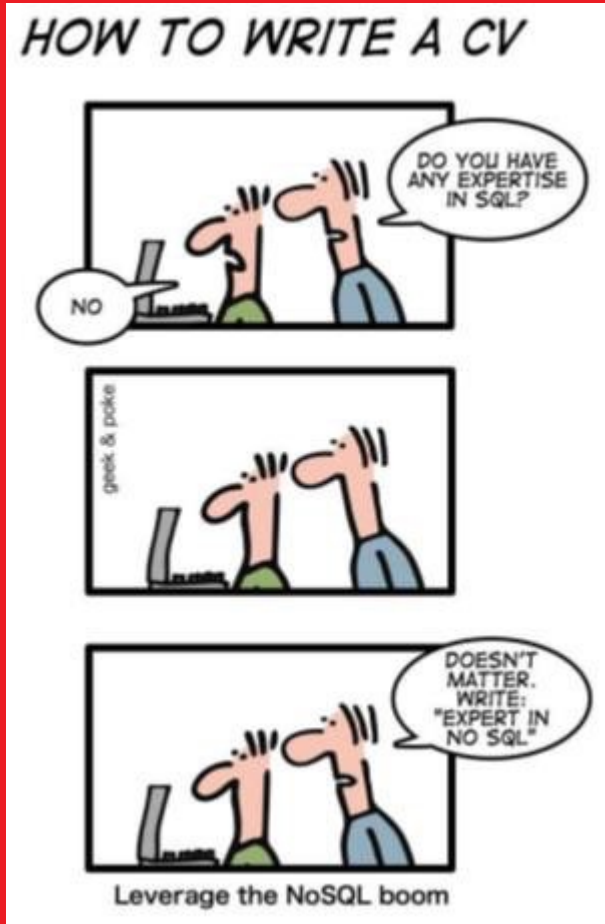
## Based on 3Vs

- **Variety**: Variations in the structure of data to be stored
  - **Structured data** fits into a predefined data model
  - **Unstructured data** does not fit into a predefined model

# Limitations of relational DBs

- **Data model**
  - A predefined data model (i.e., the schema) may not be suitable for unstructured or heterogeneous data
- **Query language**
  - The support for flexible and complex queries may not be required for specific applications
- **Distributed data integrity**
  - Relational database systems don’t scale well over distributed servers because of data partitioning and join problems.
- **Complexity**
  - Many unnecessary features for specific purposes.

# NO SQL


HOW TO WRITE A CV
DO YOU HAVE ANY EXPERTISE IN SQL?
NO
geek & poke
DOESN'T MATTER. WRITE: "EXPERT IN NO SQL"
Leverage the NoSQL boom

# NO SQL

- **NoSQL system**: a generic term to denote any modern non-relational system that, in particular, does not use SQL.
  - NoSQL = “Non SQL” or “Not only SQL”


  - Aims to provide better performance (query speed) and flexibility (ability to *change structure* and *increase size*). Also, *better reliability*.

# Characteristics of NoSQL systems

Griffith UNIVERSITY

- Flexible schema, easy to set up
- Support distributed database architectures
- Focus on massive scalability, high availability, and fault tolerance.
- Do *not* have a high-level query language
  - It's hence necessary to write applications in some lower-level programming language.

# NoSQL databases

- In general NoSQL databases can be:
  - **key-value**: SimpleDB, Redis, Memcached, Dynamo, Voldemort
  - **document**: MongoDB, CouchDB
  - **column-oriented**: BigTable, HBase, Hypertable, Cassandra, PNUTS
  - **graph**: Neo4j, GraphDB

# Key-value database

- **Key-value (KV) databases** store data as a collection of key-value pairs organized as **buckets** which are the equivalent of tables

FIGURE 14.7 KEY-VALUE DATABASE STORAGE

Bucket = Customer

| Key | Value |
|---|---|
| 10010 | "LName Ramas FName Alfred Initial A Areacode 615 Phone 844-2573 Balance 0" |
| 10011 | "LName Dunne FName Leona Initial K Areacode 713 Phone 894-1238 Balance 0" |
| 10014 | "LName Orlando FName Myron Areacode 615 Phone 222-1672 Balance 0" |

- A **bucket** corresponds roughly to a relational table, key is unique in a bucket.

# Document database

- **Document databases** store data in key-value pairs in which the value components are tag-encoded documents grouped into logical groups called **collections**

FIGURE 14.8 DOCUMENT DATABASE TAGGED FORMAT

Collection = Customer

| Key | Document |
|---|---|
| 10010 | {LName: "Ramas", FName: "Alfred", Initial: "A", Areacode: "615", Phone: "844-2573", Balance: "0"} |
| 10011 | {LName: "Dunne", FName: "Leona", Initial: "K", Areacode: "713", Phone: "894-1238", Balance: "0"} |
| 10014 | {LName: "Orlando", FName: "Myron", Areacode: "615", Phone: "222-1672", Balance: "0"} |

- A **collection** corresponds roughly to a relational table.

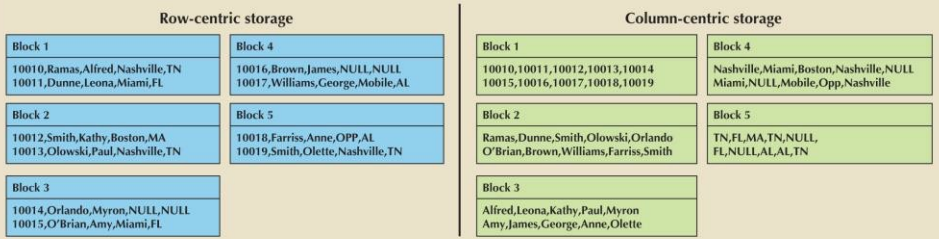
# Column-oriented database

- **Column-oriented databases** refers to two technologies:
  - **Column-centric storage**: Data stored in blocks which hold data from a single column across many rows
  - **Row-centric storage:** Data stored in block which hold data from all columns of a given set of rows

FIGURE 14.9 COMPARISON OF ROW-CENTRIC AND COLUMN-CENTRIC STORAGE

CUSTOMER relational table

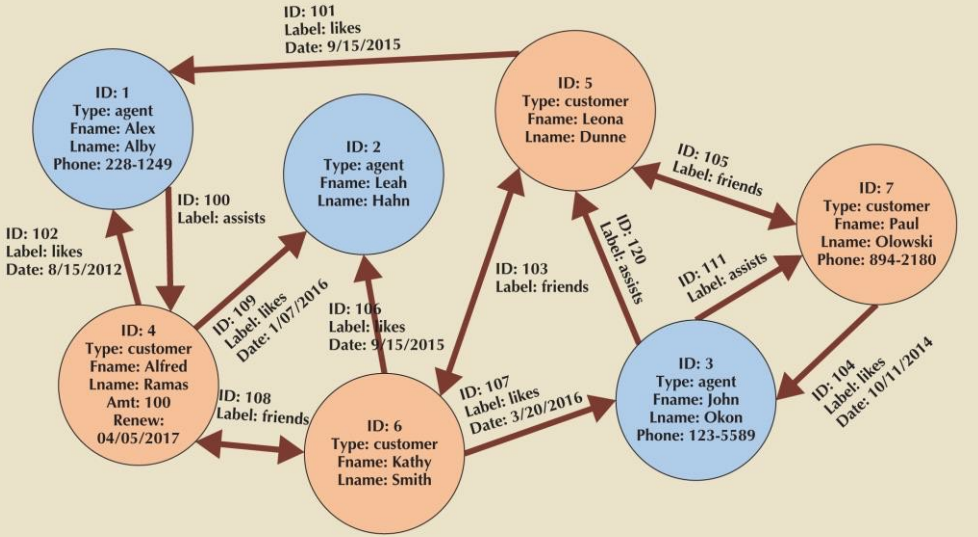
| Cus_Code | Cus_LName | Cus_FName | Cus_City | Cus_State |
|---|---|---|---|---|
| 10010 | Ramas | Alfred | Nashville | TN |
| 10011 | Dunne | Leona | Miami | FL |
| 10012 | Smith | Kathy | Boston | MA |
| 10013 | Olowski | Paul | Nashville | TN |
| 10014 | Orlando | Myron | | |
| 10015 | O'Brian | Amy | Miami | FL |
| 10016 | Brown | James | | |
| 10017 | Williams | George | Mobile | AL |
| 10018 | Farriss | Anne | Opp | AL |
| 10019 | Smith | Olette | Nashville | TN |



# Graph database

- **Graph databases** store data on relationship-rich data as a collection of **nodes** and **edges**

FIGURE 14.11 GRAPH DATABASE REPRESENTATION




- Nodes may have **properties**, which are the attributes of a node of interest to a user (including ID)
- Edge may have **labels** or **roles**
- **Traversal** is a query in a graph database

# Example: Watch Amazon DynamoDB



- Go to https://aws.amazon.com/nosql/
- YouTube link: https://www.youtube.com/watch?v=sI-zciHAh-4


- See more:
  - https://aws.amazon.com/dynamodb/

# Have a question?