Lecture notes

CDS 302

Scientific Data and Databases

Fall Semester 2020

Lecture 1: Introduction

Lecture: Joe Boone

Who am I?

- Joe Boone – GMU Graduate Lecturer
  - jboone@gmu.edu

- Academic Background
  - BS and MS in Computer Science from GMU
  - Currently a Computational Science and Informatics Ph.D. Student at GMU

- Professional Career
  - 30+ years of Systems Development and Engineering
  - Satellite Telecommunication Applications
  - Geospatial Applications
  - Extensive Software Development Experience
  - Graduate Lecturer at GMU

Week 1 Topics

- Syllabus Review
- Tools
- Introduction to Database Systems
- Introduction to Scientific Writing
- Introduction to LaTeX and Overleaf (Part 1)
- Week 1 Assignments

Syllabus Review

Tools

Software You Will Need

www.overleaf.com
www.sqlite.org/download.html
www.sqlitebrowser.org/dl/

www.anaconda.com/distribution/

An Introduction to Database Systems

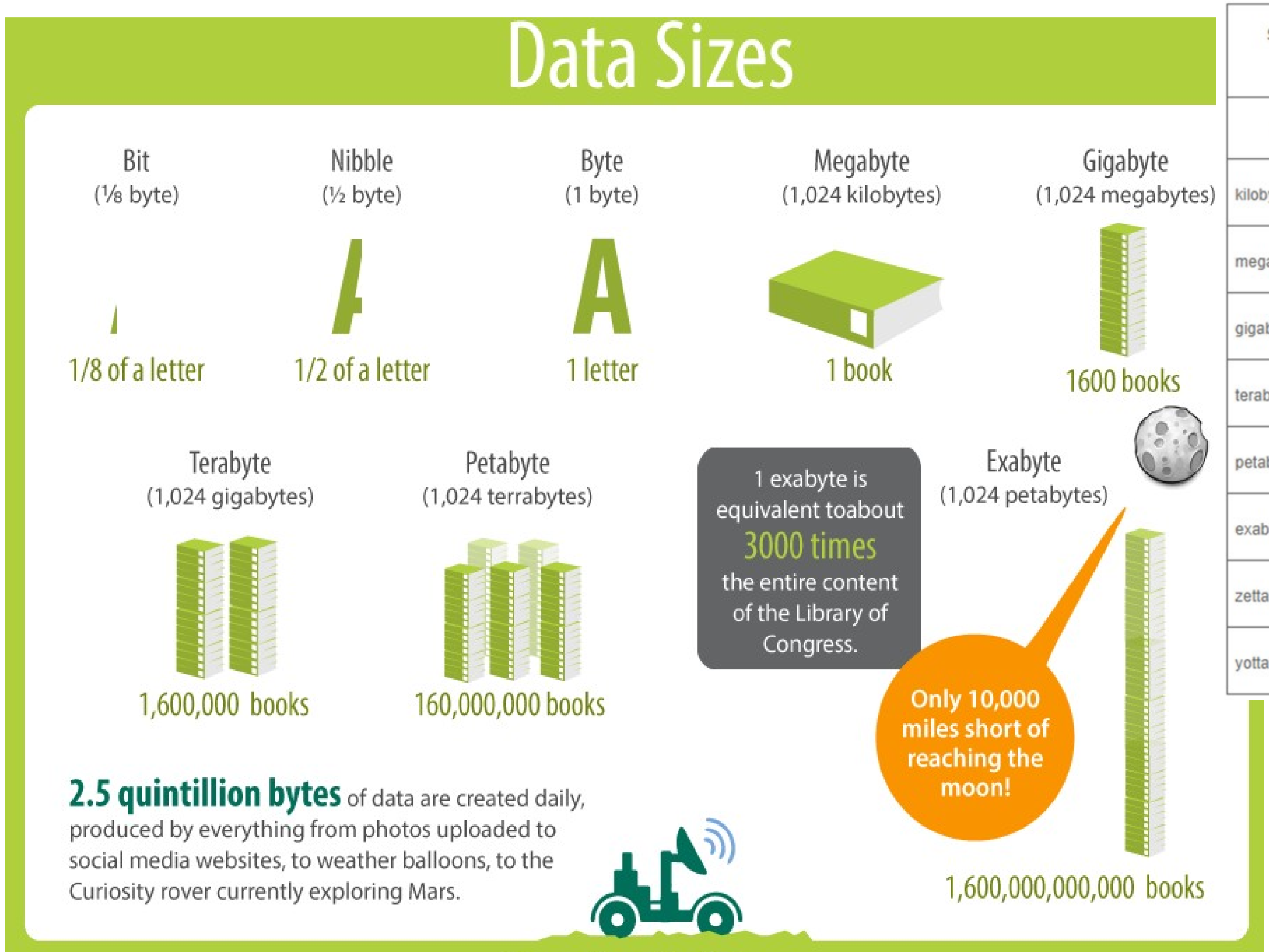
Some Basic Terminology

- **Database**
  - Database: Data collection, typically large and stored in secondary memory
  - Contains interrelated data on some enterprise

- **Database Management System (DBMS)**
  - Database Management System: a software package (i.e., collection of programs) designed to *define, manipulate, retrieve* and *manage* data in a database
  - Provides an environment that is convenient and efficient to use for multiple users and applications simultaneously

- **Database System**
  - DBMS + Database

Data Sizes Reference



Data Sizes

Bit (⅛ byte) — 1/8 of a letter

Nibble (½ byte) — 1/2 of a letter

Byte (1 byte) — 1 letter

Megabyte (1,024 kilobytes) — 1 book

Gigabyte (1,024 megabytes) — 1600 books

Terabyte (1,024 gigabytes) — 1,600,000 books

Petabyte (1,024 terrabytes) — 160,000,000 books

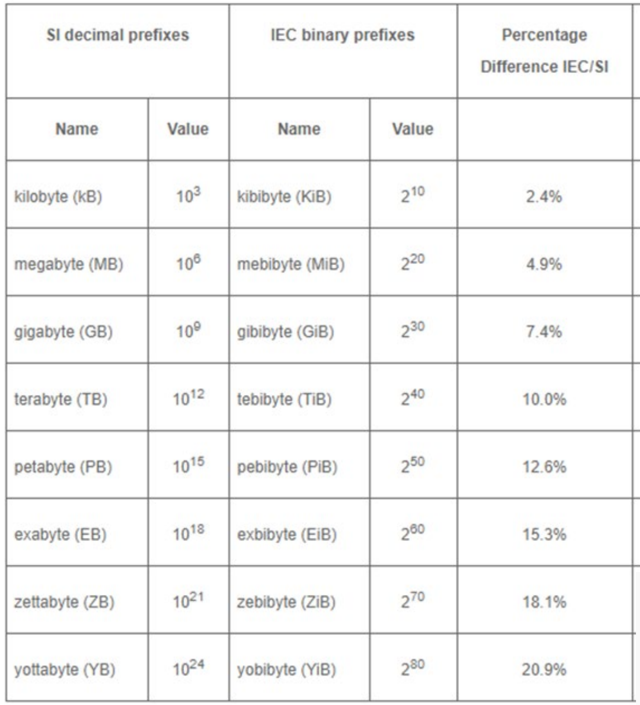
1 exabyte is equivalent toabout 3000 times the entire content of the Library of Congress.

Exabyte (1,024 petabytes) — 1,600,000,000,000 books

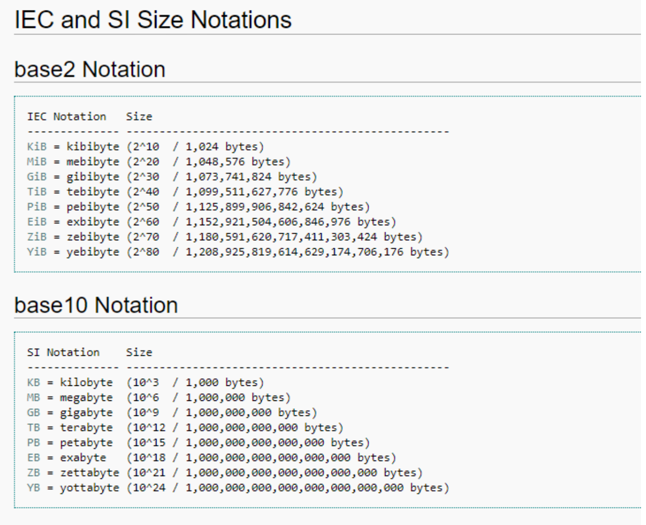
Only 10,000 miles short of reaching the moon!

2.5 quintillion bytes of data are created daily, produced by everything from photos uploaded to social media websites, to weather balloons, to the Curiosity rover currently exploring Mars.

Bigger Than Big Data



SI decimal prefixes		IEC binary prefixes		Percentage Difference IEC/SI
Name	Value	Name	Value	
kilobyte (kB)	10^3	kibibyte (KiB)	2^{10}	2.4%
megabyte (MB)	10^6	mebibyte (MiB)	2^{20}	4.9%
gigabyte (GB)	10^9	gibibyte (GiB)	2^{30}	7.4%
terabyte (TB)	10^{12}	tebibyte (TiB)	2^{40}	10.0%
petabyte (PB)	10^{15}	pebibyte (PiB)	2^{50}	12.6%
exabyte (EB)	10^{18}	exbibyte (EiB)	2^{60}	15.3%
zettabyte (ZB)	10^{21}	zebibyte (ZiB)	2^{70}	18.1%
yottabyte (YB)	10^{24}	yobibyte (YiB)	2^{80}	20.9%



IEC and SI Size Notations

base2 Notation

```
IEC Notation   Size
-------------- ------------------------------------------------
KiB = kibibyte (2^10  / 1,024 bytes)
MiB = mebibyte (2^20  / 1,048,576 bytes)
GiB = gibibyte (2^30  / 1,073,741,824 bytes)
TiB = tebibyte (2^40  / 1,099,511,627,776 bytes)
PiB = pebibyte (2^50  / 1,125,899,906,842,624 bytes)
EiB = exbibyte (2^60  / 1,152,921,504,606,846,976 bytes)
ZiB = zebibyte (2^70  / 1,180,591,620,717,411,303,424 bytes)
YiB = yebibyte (2^80  / 1,208,925,819,614,629,174,706,176 bytes)
```

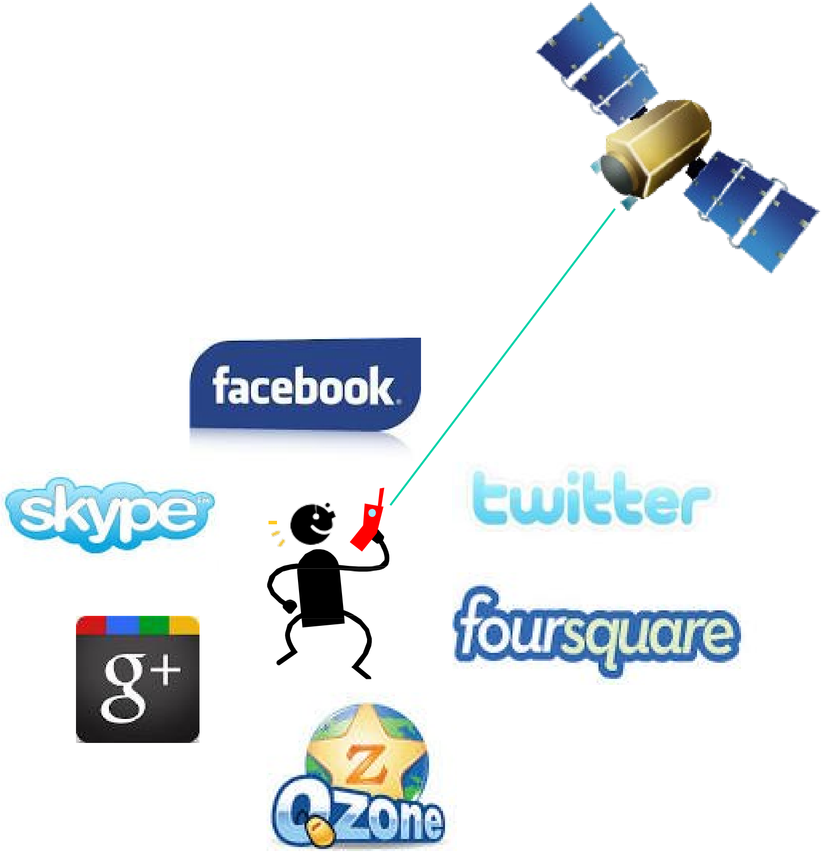
base10 Notation

```
SI Notation    Size
-------------- ------------------------------------------------
KB = kilobyte  (10^3  / 1,000 bytes)
MB = megabyte  (10^6  / 1,000,000 bytes)
GB = gigabyte  (10^9  / 1,000,000,000 bytes)
TB = terabyte  (10^12 / 1,000,000,000,000 bytes)
PB = petabyte  (10^15 / 1,000,000,000,000,000 bytes)
EB = exabyte   (10^18 / 1,000,000,000,000,000,000 bytes)
ZB = zettabyte (10^21 / 1,000,000,000,000,000,000,000 bytes)
YB = yottabyte (10^24 / 1,000,000,000,000,000,000,000,000 bytes)
```

Why do we need Database Systems?
Data is Everywhere

- Huge flood of data
  - Modern technology
  - New user mentality
  - 2.5 Exabytes of new data every day

- New applications
- Innovative research
- Economic Boost
  - “$600 billion potential annual consumer surplus from using personal location data” **[1]**



[1] McKinsey Global Institute. Big data: The next frontier for innovation, competition, and productivity. June 2011.

Why do we need Database Systems?
Data Production is Accelerating

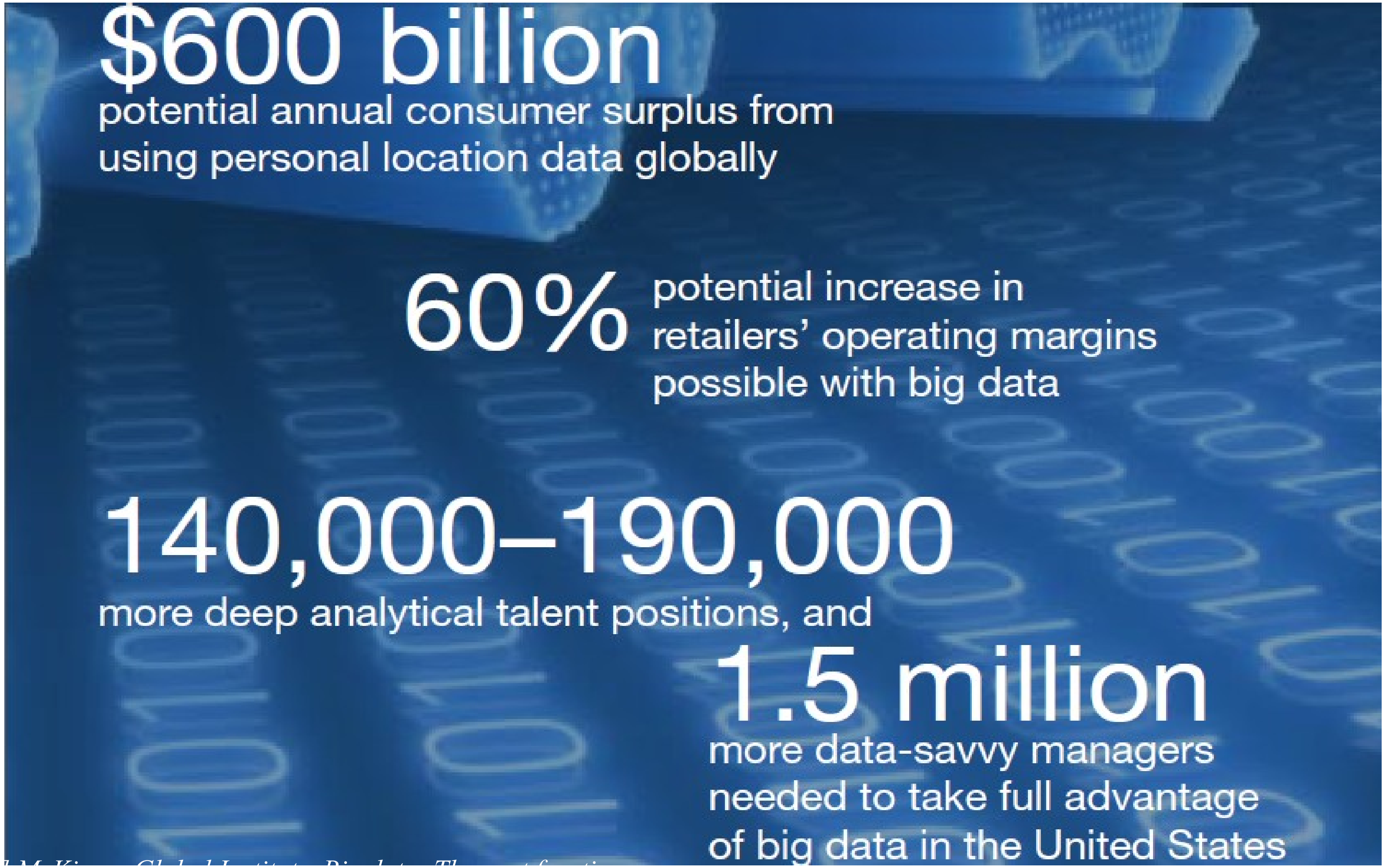
From the dawn of civilization until 2003, humankind generated five exabytes of data. Now we produce five exabytes every two days…and the pace is accelerating.

Eric Schmidt,
Executive Chairman, Google

Quoted in 2010…

Why do we need Database Systems?
Big Data = $$$

Opportunity…



$600 billion
potential annual consumer surplus from using personal location data globally

60% potential increase in retailers’ operating margins possible with big data

140,000–190,000
more deep analytical talent positions, and

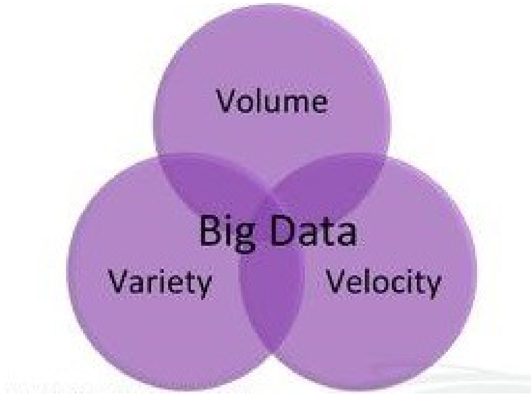
1.5 million
more data-savvy managers needed to take full advantage of big data in the United States

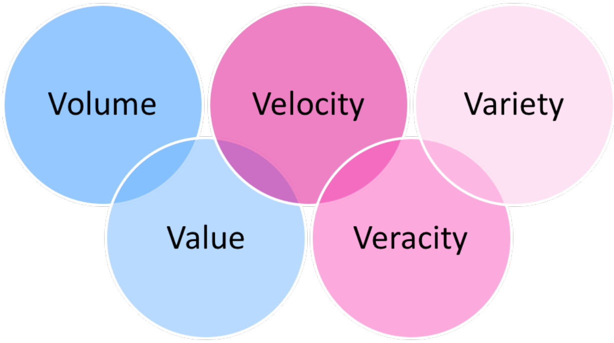
McKinsey Global Institute, Big data: The next frontier

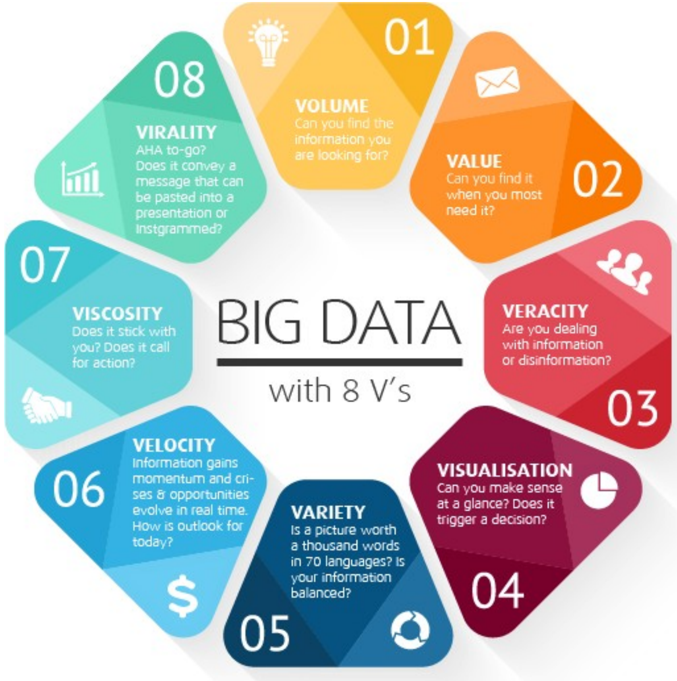
Why do we need Database Systems?
Big Data Hype

The 3-*ish* V's of Big data…

- Volume
- Velocity
- Variety
    - ○ *Veracity*
    - ○ *Variability*
    - ○ *Visualization*
    - ○ *Value*
    - …







VOLUME Can you find the information you are looking for?

VALUE Can you find it when you most need it?

VERACITY Are you dealing with information or disinformation?

VISUALISATION Can you make sense at a glance? Does it trigger a decision?

VARIETY Is a picture worth a thousand words in 70 languages? Is your information balanced?

VELOCITY Information gains momentum and crises & opportunities evolve in real time. How is outlook for today?

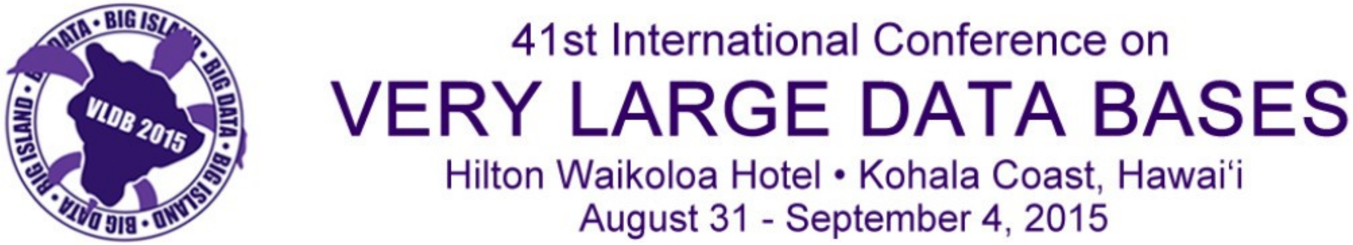
VISCOSITY Does it stick with you? Does it call for action?

VIRALITY AHA to-go? Does it convey a message that can be pasted into a presentation or Instgrammed?

Why do we need Database Systems?
Big Data Hype

- Big Data analytics is a fancy new word for Knowledge Discovery in Databases!

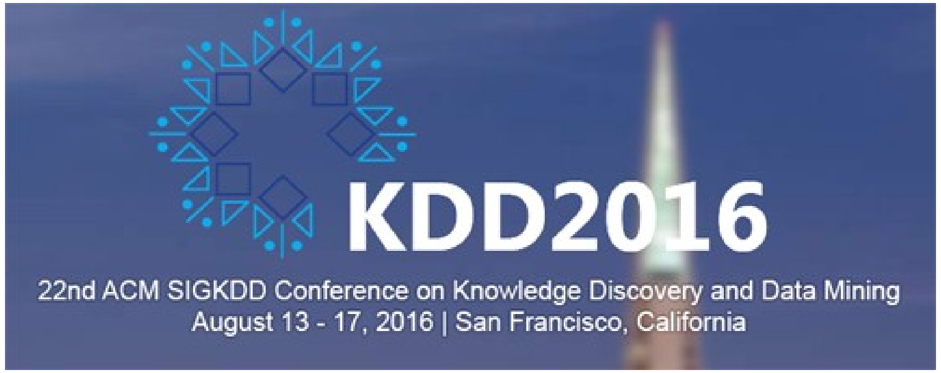
- KDD has focused on large data for decades:


BIG ISLAND • BIG DATA • BIG ISLAND • BIG DATA • BIG ISLAND • BIG DATA
VLDB 2015

41st International Conference on
VERY LARGE DATA BASES
Hilton Waikoloa Hotel • Kohala Coast, Hawai'i
August 31 - September 4, 2015

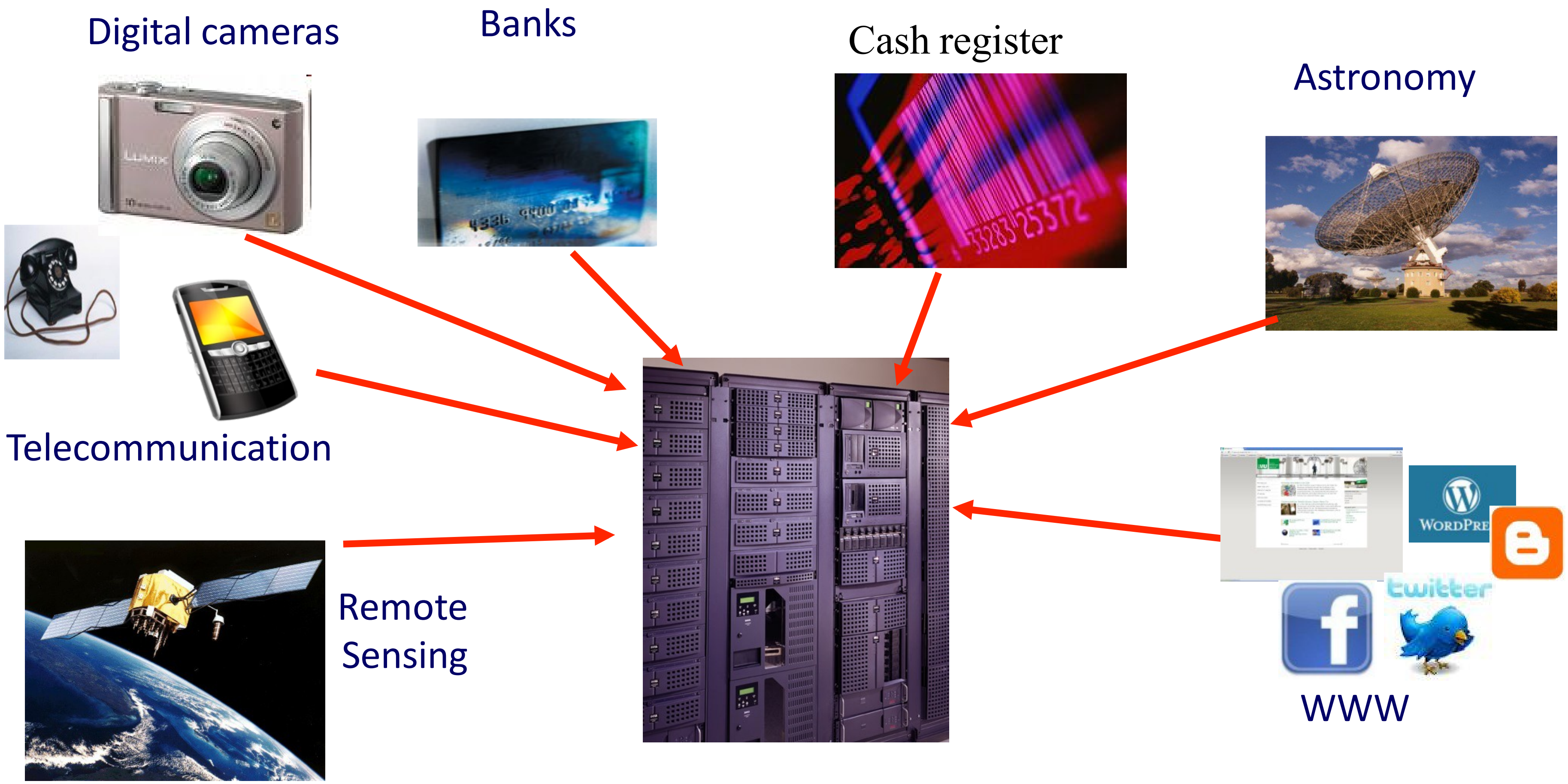
  - **VL**DB since 1975

- **Big Data is not new**



KDD2016
22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining
August 13 - 17, 2016 | San Francisco, California

Why do we need Database Systems? Data Collection



Digital cameras

Banks

Cash register

Astronomy

Telecommunication

Remote Sensing

WWW

- Huge amounts of data are collected nowadays from different application domains
- Is not feasible to analyze all these data manually

Why do we need Database Systems?
Who uses databases?

Zillow

G

tripadvisor

amazon

ebay

f

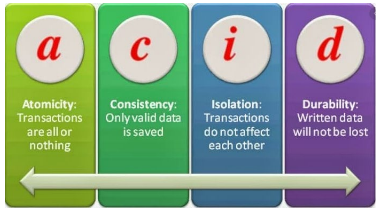
Everyone…

Slide from Katerina Doka, Introduction to Databases

Why do we need Database Systems?

- What problems do database systems solve:
  - **Data redundancy and inconsistency**
    - Duplication of data, different values for the same variable
  - **Difficulty in accessing the data**
  - **Data isolation**
    - Different files and formats complexity is hidden from users (abstracted)
  - **Data integrity**
    - Data is not logically consistent and rules to enforce are encoded in program code
    - Hard to add/modify constraints
  - **Atomicity**
    - Failures in the middle of a transaction leave the system in an inconsistent state
  - **Concurrent access by multiple users**
  - **Managing security**
    - Need to manage who has access to what data
  - **Proprietary systems lead to vendor “lock-in” and other problems**
  - **Performance and scalability**

Terminology: *ACID Transaction,* Atomicity, Consistency, Isolation, Durability


a
c
i
d
Atomicity: Transactions are all or nothing
Consistency: Only valid data is saved
Isolation: Transactions do not affect each other
Durability: Written data will not be lost

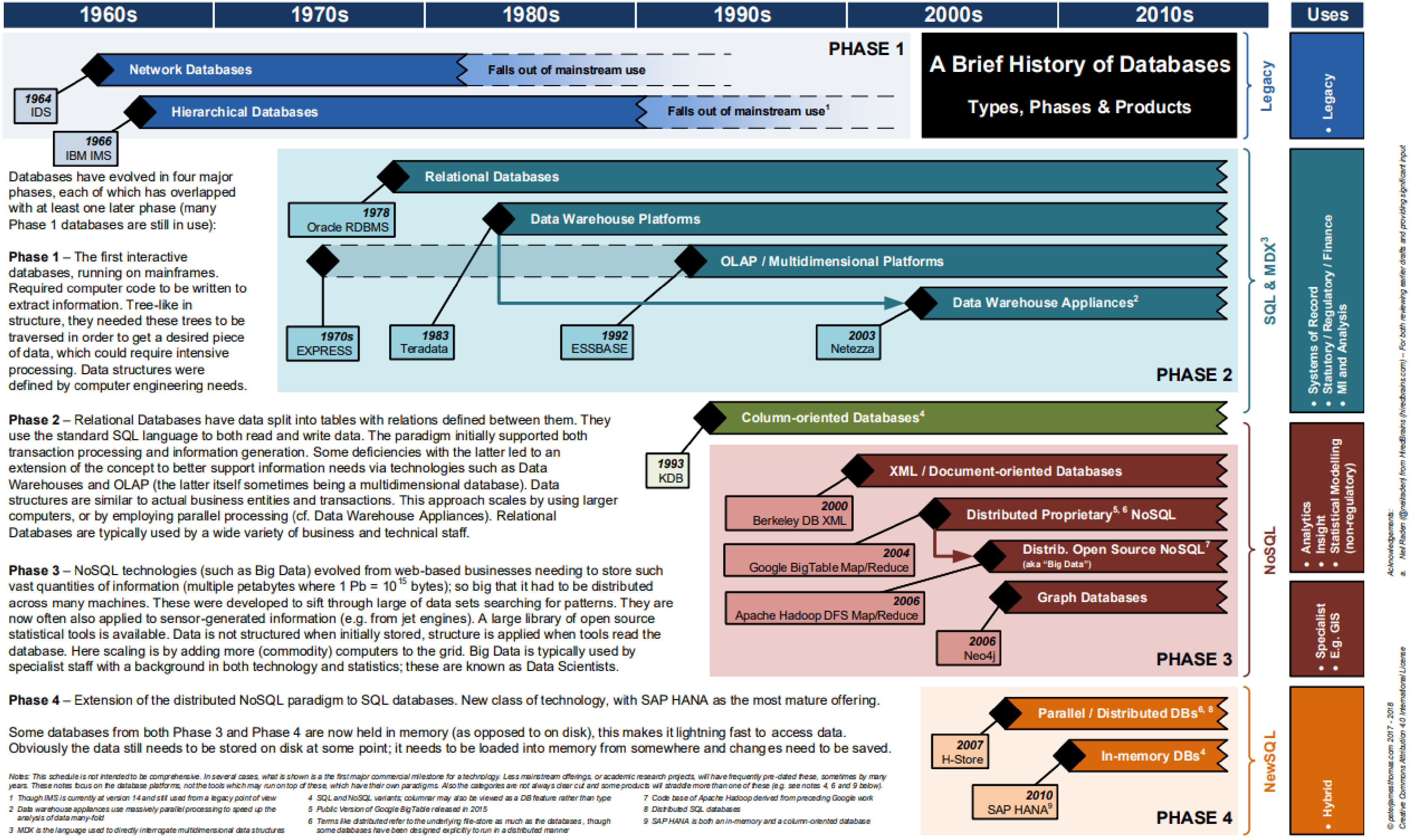
History of Database Systems

EXTREME SCIENCE

GEORGE MASON UNIVERSITY | College of Science

WHERE INNOVATION IS TRADITION

cos.gmu.edu



1960s | 1970s | 1980s | 1990s | 2000s | 2010s | Uses

PHASE 1

A Brief History of Databases

Types, Phases & Products

1964 IDS — Network Databases — Falls out of mainstream use

1966 IBM IMS — Hierarchical Databases — Falls out of mainstream use[1]

Legacy

- Legacy

PHASE 2

1978 Oracle RDBMS — Relational Databases

1983 Teradata — Data Warehouse Platforms

1970s EXPRESS; 1992 ESSBASE — OLAP / Multidimensional Platforms

2003 Netezza — Data Warehouse Appliances[2]

SQL & MDX[3]

- Systems of Record
- Statutory / Regulatory / Finance
- MI and Analysis

1993 KDB — Column-oriented Databases[4]

PHASE 3

2000 Berkeley DB XML — XML / Document-oriented Databases

2004 Google BigTable Map/Reduce — Distributed Proprietary[5, 6] NoSQL

2006 Apache Hadoop DFS Map/Reduce — Distrib. Open Source NoSQL[7] (aka "Big Data")

2006 Neo4j — Graph Databases

NoSQL

- Analytics
- Insight
- Statistical Modelling (non-regulatory)

- Specialist
- E.g. GIS

PHASE 4

2007 H-Store — Parallel / Distributed DBs[6, 8]

2010 SAP HANA[9] — In-memory DBs[4]

NewSQL

- Hybrid

Databases have evolved in four major phases, each of which has overlapped with at least one later phase (many Phase 1 databases are still in use):

Phase 1 – The first interactive databases, running on mainframes. Required computer code to be written to extract information. Tree-like in structure, they needed these trees to be traversed in order to get a desired piece of data, which could require intensive processing. Data structures were defined by computer engineering needs.

Phase 2 – Relational Databases have data split into tables with relations defined between them. They use the standard SQL language to both read and write data. The paradigm initially supported both transaction processing and information generation. Some deficiencies with the latter led to an extension of the concept to better support information needs via technologies such as Data Warehouses and OLAP (the latter itself sometimes being a multidimensional database). Data structures are similar to actual business entities and transactions. This approach scales by using larger computers, or by employing parallel processing (cf. Data Warehouse Appliances). Relational Databases are typically used by a wide variety of business and technical staff.

Phase 3 – NoSQL technologies (such as Big Data) evolved from web-based businesses needing to store such vast quantities of information (multiple petabytes where 1 Pb = 10^{15} bytes); so big that it had to be distributed across many machines. These were developed to sift through large of data sets searching for patterns. They are now often also applied to sensor-generated information (e.g. from jet engines). A large library of open source statistical tools is available. Data is not structured when initially stored, structure is applied when tools read the database. Here scaling is by adding more (commodity) computers to the grid. Big Data is typically used by specialist staff with a background in both technology and statistics; these are known as Data Scientists.

Phase 4 – Extension of the distributed NoSQL paradigm to SQL databases. New class of technology, with SAP HANA as the most mature offering.

Some databases from both Phase 3 and Phase 4 are now held in memory (as opposed to on disk), this makes it lightning fast to access data. Obviously the data still needs to be stored on disk at some point; it needs to be loaded into memory from somewhere and changes need to be saved.

Notes: This schedule is not intended to be comprehensive. In several cases, what is shown is a the first major commercial milestone for a technology. Less mainstream offerings, or academic research projects, will have frequently pre-dated these, sometimes by many years. These notes focus on the database platforms, not the tools which may run on top of these, which have their own paradigms. Also the categories are not always clear cut and some products will straddle more than one of these (e.g. see notes 4, 6 and 9 below).

[1] Though IMS is currently at version 14 and still used from a legacy point of view

[2] Data warehouse appliances use massively parallel processing to speed up the analysis of data many-fold

[3] MDX is the language used to directly interrogate multidimensional data structures

[4] SQL and NoSQL variants; columnar may also be viewed as a DB feature rather than type

[5] Public Version of Google BigTable released in 2015

[6] Terms like distributed refer to the underlying file-store as much as the databases, though some databases have been designed explicitly to run in a distributed manner

[7] Code base of Apache Hadoop derived from preceding Google work

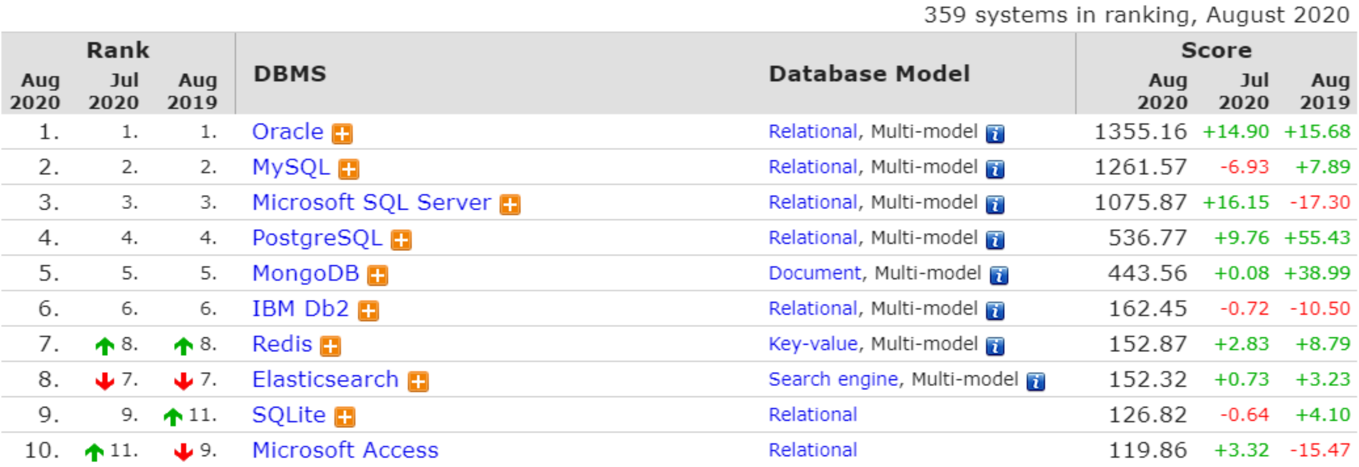
[8] Distributed SQL databases

[9] SAP HANA is both an in-memory and a column-oriented database

Acknowledgements:
a. Neil Raden (@nraden) from HiredBrains (hiredbrains.com) – For both reviewing earlier drafts and providing significant input

Slide from https://peterjamesthomas.com/2018/02/06/a-brief-history-of-databases/

Database Systems by Popularity

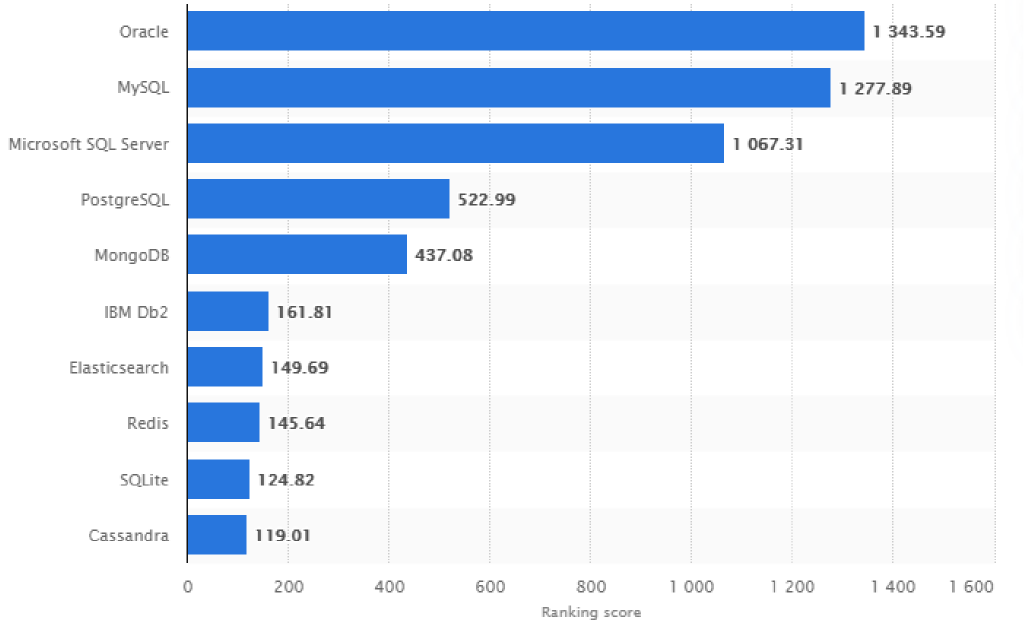


359 systems in ranking, August 2020

Rank			DBMS	Database Model	Score		
Aug 2020	Jul 2020	Aug 2019			Aug 2020	Jul 2020	Aug 2019
1.	1.	1.	Oracle	Relational, Multi-model	1355.16	+14.90	+15.68
2.	2.	2.	MySQL	Relational, Multi-model	1261.57	-6.93	+7.89
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	1075.87	+16.15	-17.30
4.	4.	4.	PostgreSQL	Relational, Multi-model	536.77	+9.76	+55.43
5.	5.	5.	MongoDB	Document, Multi-model	443.56	+0.08	+38.99
6.	6.	6.	IBM Db2	Relational, Multi-model	162.45	-0.72	-10.50
7.	↑ 8.	↑ 8.	Redis	Key-value, Multi-model	152.87	+2.83	+8.79
8.	↓ 7.	↓ 7.	Elasticsearch	Search engine, Multi-model	152.32	+0.73	+3.23
9.	9.	↑ 11.	SQLite	Relational	126.82	-0.64	+4.10
10.	↑ 11.	↓ 9.	Microsoft Access	Relational	119.86	+3.32	-15.47

https://db-engines.com/en/ranking

Relational databases are still dominant and provide a foundation to build on…



https://www.statista.com/statistics/809750/worldwide-popularity-ranking-database-management-systems/

Why Relational DBMS?

- **In our course we will be focusing on Relational Database Systems:**
  - **Data Modeling**
    - Redundancy Control
    - Referential Integrity / Consistency Constraints
  - **Efficient Query Processing**
    - Indexing
    - Optimization
  - **Operating Accuracy**
    - Error recovery - Atomicity
    - Concurrent access by multiple users
  - **Security Issues**
    - Control access rights
  - **Standards Based**
    - Structured Query Language(SQL)is an ANSI and ISO standard
    - Kind of…
  - **Large Legacy Base**
    - Lots and lots of relational databases are out there…

Why not Relational DBMS?

- Relational DBMS are designed for <u>structured</u> design and development
  - Rigidly enforced rules for data integrity
  - This is not applicable for all situations

- In modern situations, valuable data is often generated in a relatively <u>unstructured</u> format:
  - Examples: text, log files, documents, BLOBS(binary large objects) like pictures, videos, audio, etc…
  - While these can sometimes be held within a relational database – these data do not neatly fall into the relational model
  - The system being designed not require inherent Relational DBMS features: e.g. Atomicity, data integrity checking, or may have asymmetric requirements between storing and retrieving functions

Types of Database Systems?

- **Relational (Classic Relational)**
  - Oracle, IBM/DB2, MS SQL Server
  - PostgreSQL, MySQL, SQLite
- **Non-Relational (NoSQL "Not Only SQL")**
  - Key-Value Store (e.g. AWS DynamoDB, redis)
  - Document Store (e.g. MongoDB)
  - Wide-Column Stores (examples: Bigtable, Cassandra, Apache Hbase)
  - Graph Database (e.g. Neo4j)
- **Object-Oriented Databases**
  - Object-Oriented Store (e.g. InterSystems Cache)
- **Geographic Databases**
  - Vectors/Raster Store (e.g. ESRI Geodatabases / SpatiaLite)
- **First Databases…(1960s)**
  - Network (IDS, Many-to-many relationship)
  - Hierarchical (IBM IMS, Parent-child relationship)
- **DBMS as a Service (Not really a type of database…)**
  - Google
  - AWS

DBMS Architecture (briefly…)

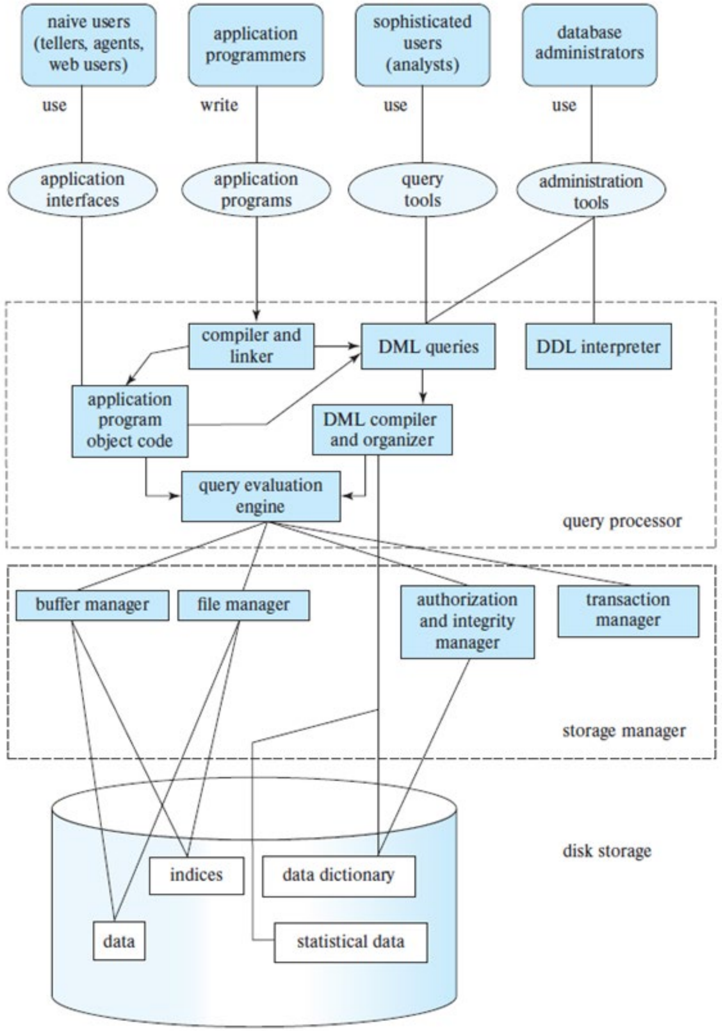


Figure 1.3 System structure.

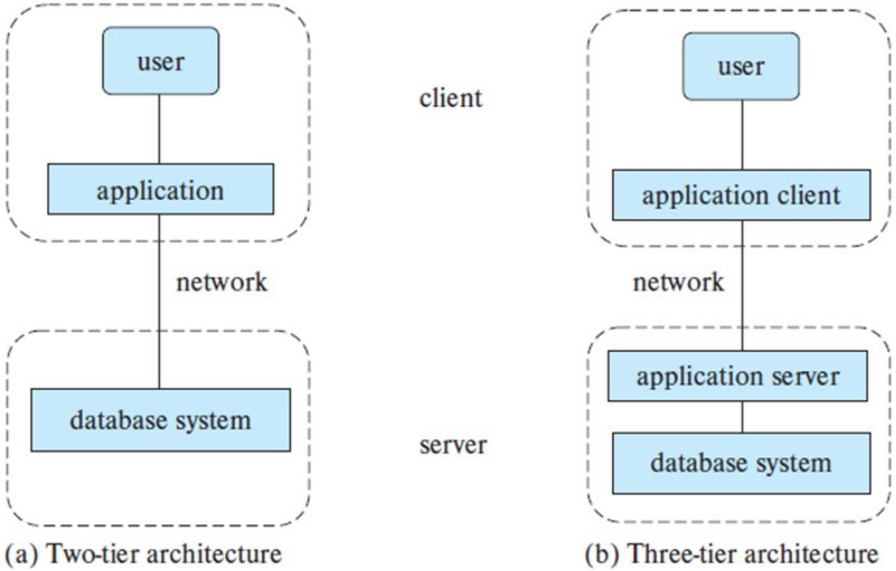


Figure 1.4 Two-tier and three-tier architectures.

From: Silberschatz et. al., Database System Concepts, McGraw-Hill

Why do I want to learn it?

DATA

Data Scientist: The Sexiest Job of the 21st Century [1]

by Thomas H. Davenport and D.J. Patil

If this means having rare qualities that are much in demand, data scientists are already there. They are difficult and expensive to hire and, given the very competitive market for their services, difficult to retain.

[1] Harvard Business Review. Data Scientist: The Sexiest Job of the 21st Century. October 2012.

An Introduction to Scientific Writing

Why read journal articles?

- To stay current with the progress in a field of study
- To find the solution for a specific problem
  - Test / Methods
- To understand the fundamental background in an area of study
- To get an idea for carrying out further research
- You have been assigned to review the article by a Professor or Journal Editor
- To support, refine, refute your scientific beliefs or views
- To impress others…

Adapted from: How to read clinical journals: I. why to read them and how to start reading them critically. Can Med Assoc J. 1981 Mar 1; 124(5):555-8; Durbin CG., Jr How to read a scientific research paper. Respir Care. 2009;54:1366–71.

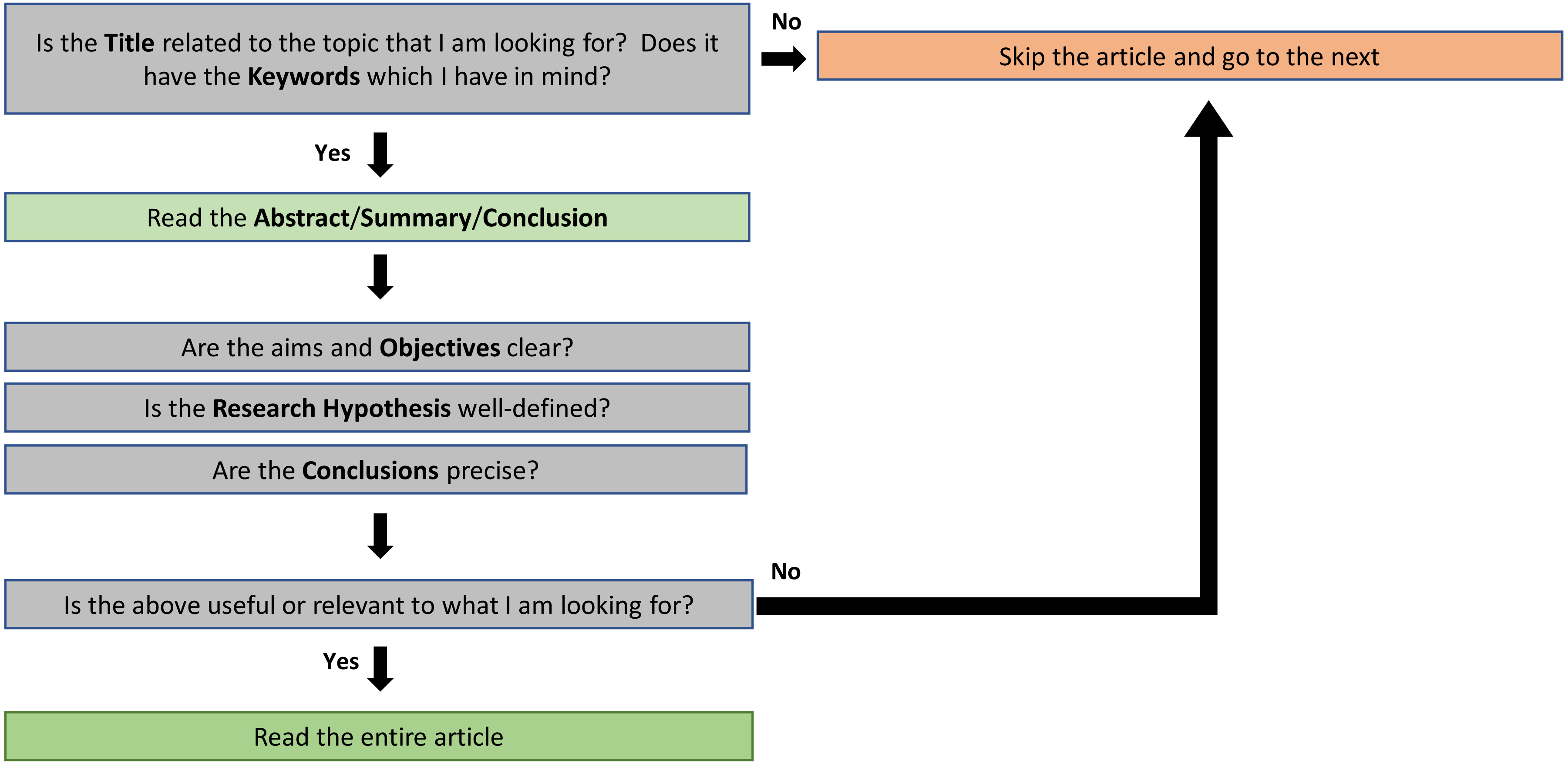
Types of articles published in a scientific journal

- **Primary literature**
  - "Core" of scientific publications
  - Present findings on new scientific discoveries
  - Describe earlier work to acknowledge it and place new findings in the proper perspective
    - Original research articles
    - Surveys
    - Case report/case series
    - Conference proceedings and abstracts
    - Editorial
    - Correspondence/letters to the Editor
- **Secondary literature**
  - Original research information is reviewed
    - Narrative reviews
    - Systematic reviews
    - Meta-analysis
    - Book reviews
    - Guidelines
    - Commentary

Structure of a journal article

- **Title:** Topic and information about the authors.
- **Abstract:** Brief overview of the article.
- **Introduction:** Background information, gap in research and statement of the research hypothesis. Include motivation – why is this important.
- **Methods:** Details on how the study was conducted, procedures followed, instruments used, and variables measured. Must be systematic.
- **Results/Experimental Evaluation:** All the data of the study along with figures, tables and/or graphs.
- **Discussion:** The interpretation of the results and implications of the study. Were the objectives met? Limitations and future work.
- **Conclusion:** What does all this mean?
- **References/Bibliography:** Citations of sources from where the information was retained.

Reading (Filtering) process



Is the **Title** related to the topic that I am looking for? Does it have the **Keywords** which I have in mind?

No

Skip the article and go to the next

Yes

Read the **Abstract/Summary/Conclusion**

Are the aims and **Objectives** clear?

Is the **Research Hypothesis** well-defined?

Are the **Conclusions** precise?

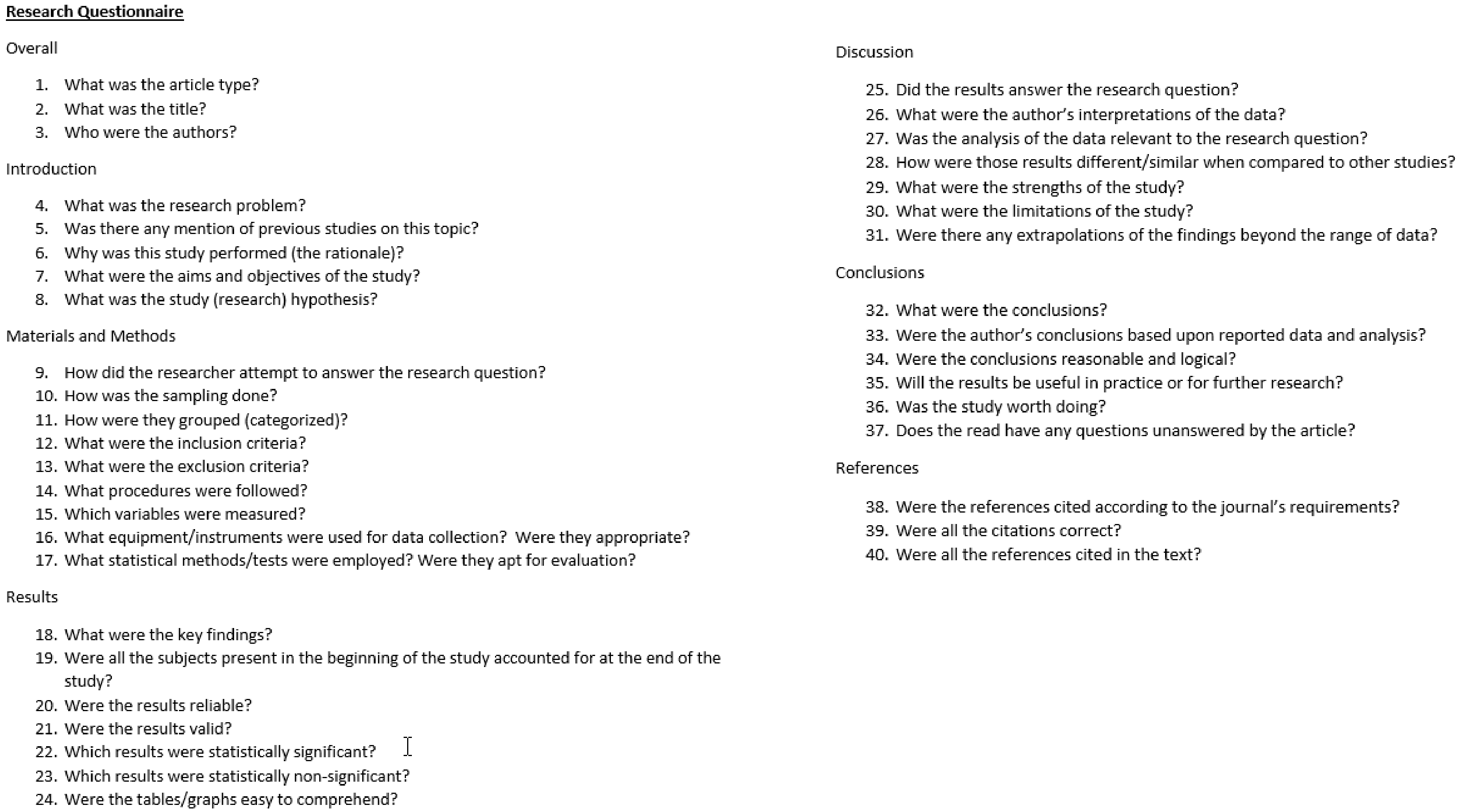
Is the above useful or relevant to what I am looking for?

No

Yes

Read the entire article

Research Questionnaire



Research Questionnaire

Overall

1. What was the article type?
2. What was the title?
3. Who were the authors?

Introduction

4. What was the research problem?
5. Was there any mention of previous studies on this topic?
6. Why was this study performed (the rationale)?
7. What were the aims and objectives of the study?
8. What was the study (research) hypothesis?

Materials and Methods

9. How did the researcher attempt to answer the research question?
10. How was the sampling done?
11. How were they grouped (categorized)?
12. What were the inclusion criteria?
13. What were the exclusion criteria?
14. What procedures were followed?
15. Which variables were measured?
16. What equipment/instruments were used for data collection? Were they appropriate?
17. What statistical methods/tests were employed? Were they apt for evaluation?

Results

18. What were the key findings?
19. Were all the subjects present in the beginning of the study accounted for at the end of the study?
20. Were the results reliable?
21. Were the results valid?
22. Which results were statistically significant?
23. Which results were statistically non-significant?
24. Were the tables/graphs easy to comprehend?

Discussion

25. Did the results answer the research question?
26. What were the author's interpretations of the data?
27. Was the analysis of the data relevant to the research question?
28. How were those results different/similar when compared to other studies?
29. What were the strengths of the study?
30. What were the limitations of the study?
31. Were there any extrapolations of the findings beyond the range of data?

Conclusions

32. What were the conclusions?
33. Were the author's conclusions based upon reported data and analysis?
34. Were the conclusions reasonable and logical?
35. Will the results be useful in practice or for further research?
36. Was the study worth doing?
37. Does the read have any questions unanswered by the article?

References

38. Were the references cited according to the journal's requirements?
39. Were all the citations correct?
40. Were all the references cited in the text?

It takes time…

- It the not the same as reading a novel or a blog
- It’s a skill
- It gets better with practice
- It gets better as you become more familiar with the research area
- The first paper may take some time
- You may have to look at other resources to understand some of the paper’s content
- Be patient....you’ll get there!

Additional resources

- Google Search
- Google Scholar
- GMU Library resources (library.gmu.edu)
- Academia.edu
- ResearchGate
- Reddit Scholar
- Email scholars if you can’t get their articles freely

- Citations managers like Zotero or Mendeley
  - Help you keep your research materials organized and in one place
  - Help you generate a bibliography

Introduction to $\LaTeX$

LaTeX

- LaTeX
  - LaTeX is “Lamport” + “TeX”
  - Leslie Lamport
    - Computer Scientist, Distributed Systems (2013 Turing Award)
    - Initial Developer / Inventor of LaTeX (1983)
    - LaTeX is a tool for document preparation built on top of the typesetting system TeX
    - It is the standard for scientific journal articles
    - Pronounced (“la” or “lay”) + “tech”

  - TeX
    - Late 1970’s
    - Donald Knuth, Computer Scientist (Turing Award 1974, many awards…)
    - Author of *The Art of Computer Programming* (classic CS text)
    - Typesetting engine that drives LaTeX and other higher-level packages

Assignments

Assignments Week 1

- Setup a free account with Overleaf (www.overleaf.com)
  - Follow along with the recorded lectures covering LaTeX
  - Explore LaTeX on your own…(Next week we will cover Part 2)

- Suggested reading: Silberschatz et. al., *Database System Concepts, McGraw-Hill,* Chapter 1

Other Resources

- *Introduction to Databases* class by Jennifer Widom, Stanford
  - http://www.db-class.org/course/auth/welcome

- LaTeX tutorials
  - https://www.overleaf.com/learn/latex/Tutorials
  - https://www.latex-tutorial.com/tutorials/

- List of LaTeX Math Symbols
  - https://www.caam.rice.edu/~heinken/latex/symbols.pdf