

DB Class Notes

Followup for CSC330 Database Theory & Applications Spring 2022

README

Instead of bugging you with emails, I opt to summarize my course observations regarding content, process, in this file. These often contain additional links, articles, and musings.

I usually update it after each class - it also contains the homework (if any). The first point of call for any questions should be the FAQ. There are two FAQs - a [general one](#) (for all my courses), and a [FAQ for CSC 330](#).

You find the whiteboard photos [here in GDrive](#).

The companion file to this file, with the agenda and much of the course content, is the [agenda.org](#) file.

Intro and GitHub - w1s1 (01/11/22)

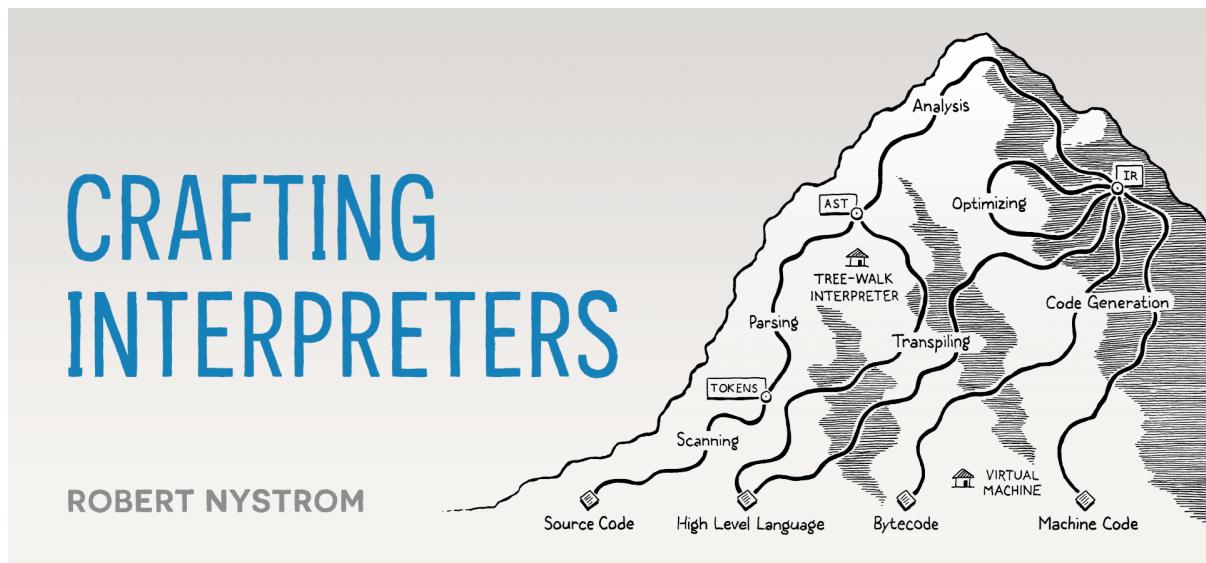
Homework (by Thursday 13-Jan)

- Register with GitHub (2 min)
- Complete the GitHub Hello World exercise (20 min)
- ~~Give me your GitHub user name (email - with course ID - or Thursday in class) - so that I can add you to the repo - no need, course is public now~~
- Submit an issue to the course repo confirming that you completed the exercise.

Note: if you already did this as part of another course, you don't have to do it again but you need to let me know in your email, which of my courses you attend so that I can add you to one or more GitHub repos.

Stuff

- [Agenda](#) - we covered all of it (and more) - agenda is available in GitHub only.
- Books: "Crafting Interpreters" by Nystrom (2021) is a fun book on creating a small interpreted programming language. This would be an extra cool (Honors) project in a course on programming languages!



- Pyret (thank you, Molly!) - [check it out](#).
- Showed GitHub, DataCamp teams and assignments, and GNU Emacs. We talked about structured vs. unstructured data, and Torvald's Git version control program: which follows a key-value data model - all data (committed differences) are stored in tree-like structures and indexed by hash codes. [See here for more](#) (Spajic, 2018).

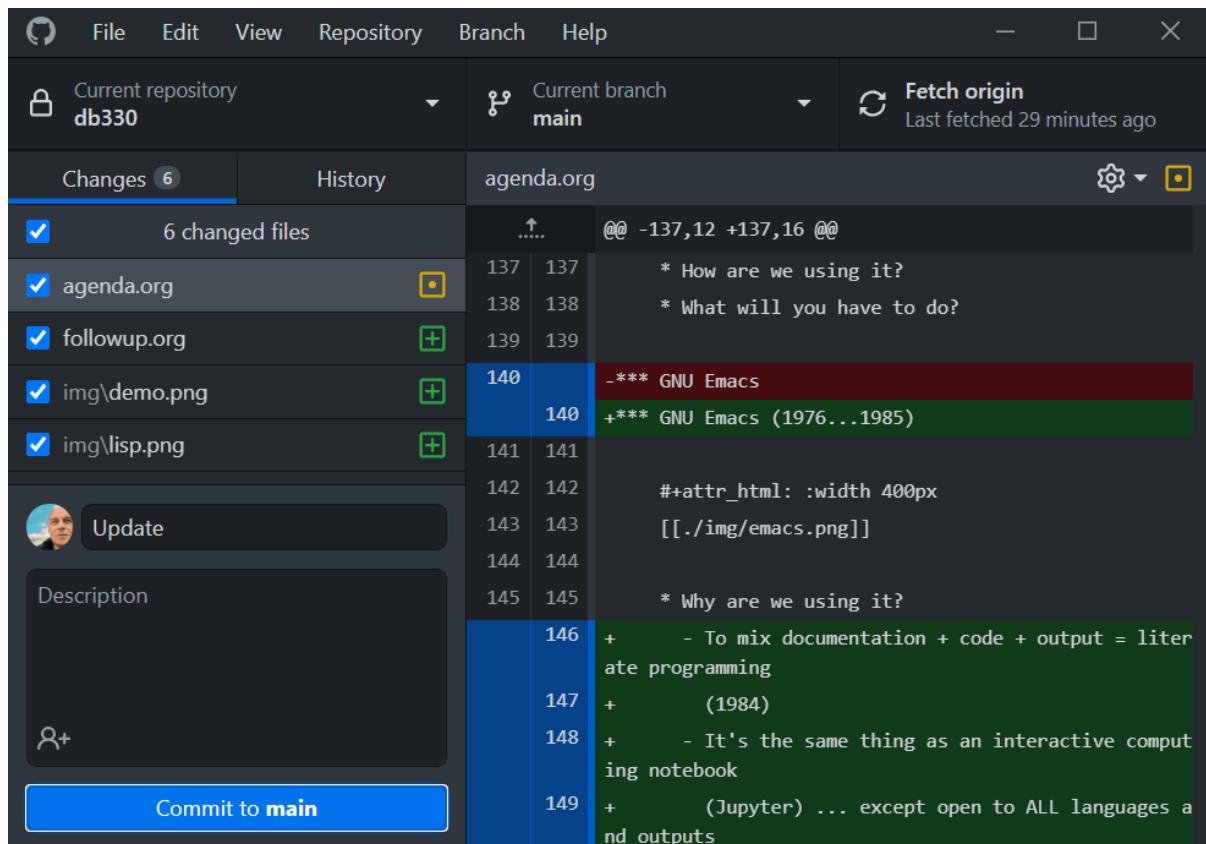


Figure 2: GitHub desktop application

- The DataCamp platform supports the three most important data science languages (in order of importance in the real world): SQL, R, Python. You can do data science with the shell (bash), or even with Java, C, or any other language of course. The season's last assignment is a project where you can see all that you learnt about SQL at work: an analysis of international debt statistics. You do not need to work on the assignments in any particular order - as long as you don't miss the weekly deadlines. A weekly assignment should take 20-30 minutes of your time.

The screenshot shows a DataCamp project interface. On the left, there is a sidebar titled 'Project Instructions' with a numbered list of tasks:

- Task 1: Instructions**
- Inspect the international debt data.
- Read the line of code provided for you, which connects you to the `international_debt` database.
- Select all of the columns from the `international_debt` table and limit the output to the first 10 rows.
- Good to know**
- The only prerequisite to complete this project is familiarity with the contents covered in DataCamp's [Introduction to SQL](#) course.
- SQL DataCamp projects are completed in Jupyter Notebooks. If you're not familiar with Jupyter Notebooks, that's okay! All you need to know is that you can execute SQL commands in the code cells provided, as long as you

Below the sidebar is a Jupyter notebook interface with a toolbar and a code cell containing the following text:

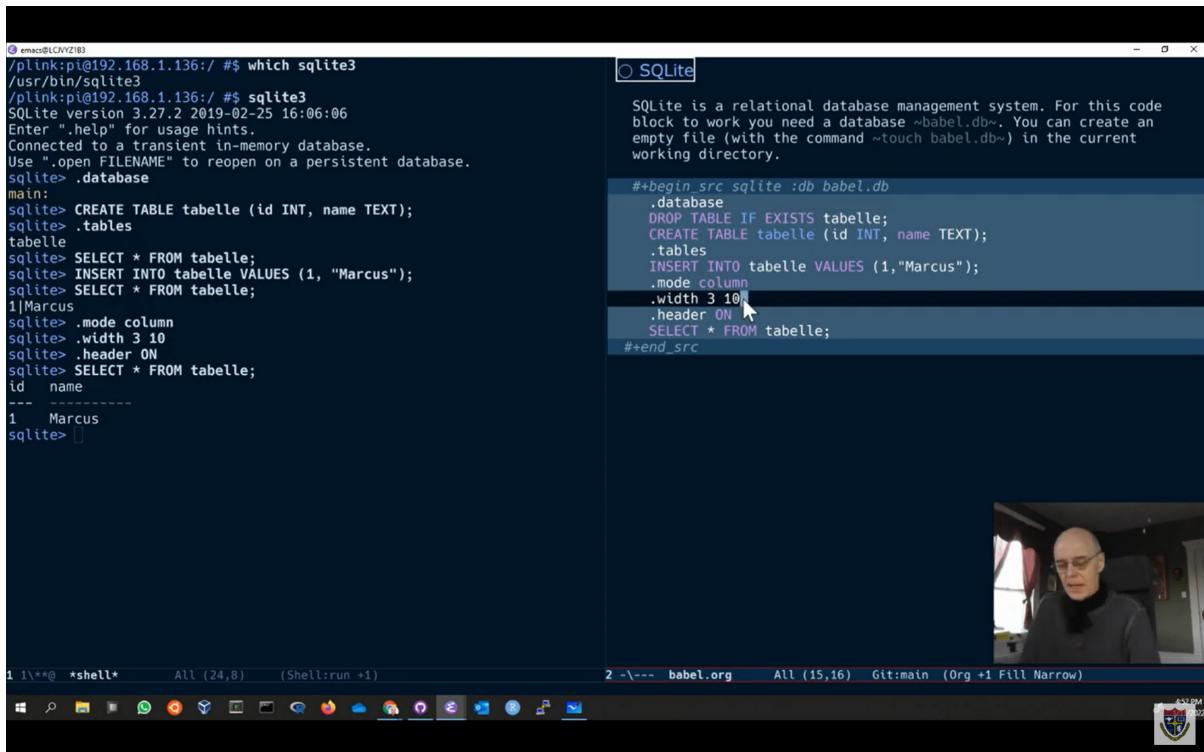
```
1. The World Bank's international debt data
It's not that we humans only take debts to manage our necessities. A country may also take debt to manage its economy. For example, infrastructure spending is one costly ingredient required for a country's citizens to lead comfortable lives. The World Bank is the organization that provides debt to countries.

In this notebook, we are going to analyze international debt data collected by The World Bank. The dataset contains information about the amount of debt (in USD) owed by developing countries across several categories. We are going to find the answers to questions like:
• What is the total amount of debt that is owed by the countries listed in the dataset?
• Which country owns the maximum amount of debt and what does that amount look like?
• What is the average amount of debt owed by countries across different debt indicators?
```

On the right side of the notebook, there is an image of two US dollar bills. Below the image, the caption reads: "The first line of code connects us to the `international_debt` database where the table". At the bottom right of the notebook area, there are 'Check Project' and 'Save' buttons.

Figure 3: DataCamp project - Analyze Intern. Debt Stats

- Recorded a screencast for the missing 10 minutes at the end (network outage in Batesville!) - [see on YouTube here](#) (Birkenkrahe, 2022). This is a demonstration of the differences of interactive shell scripting vs. interactive notebooks, using SQLite as a sample language. You will get an assignment to create a literate programming file next week. Don't be put off by the level of detail of this demo - let it all just wash over you for now - you'll learn this and much more in this course!



In the screencast I talk about missing syntax highlighting in the shell buffer on the left: "I could add it if I wanted to." This is true - everything in Emacs is customizable, and I spent years, literally, doing this. [Here is the documentation](#) specifically on customizing the shell, if you're curious. The language to do this is Emacs-Lisp, a Lisp dialect. I mentioned that Lisp was the [first and, for a long time, dominant AI language](#) (Valencia, 2017). It's a great language to learn, and GNU Emacs is the ticket if you feel like it.

```
(defun make-perceptron (n m &optional (g #'(lambda (i) (step-function 0 i)))
  &aux (l nil))

  (dotimes (i m (list l))
    (push (make-unit :parents (iota (1+ n))
      :children nil
      :weights (random-weights (1+ n) -0.5 +0.5)
      :g g)
      l)))
```

Figure 5: Common Lisp code to create an n-inputs m-units one layer perceptron. Source: AIMA.

DataCamp, History of DB, MooCall - w1s2 (01/13/22)

DataCamp membership

- DataCamp: You should all be in your courses now.
 - Your assignments are on one page but you'll be notified via schoology as soon as an assignment is due

<input type="checkbox"/>	 A	Aisha Mahmoud aisha.mahmoud@lyon.edu	CSC330 DATABASE THEORY AND APP... +1	Member	
<input type="checkbox"/>	 B	Andrei Galca andreicri.galcamitr@lyon.edu	CSC330 DATABASE THEORY AND APP... +1	Member	
<input type="checkbox"/>	 C	Blake Huffman blake.huffman@lyon.edu	CSC330 DATABASE THEORY AND APP... +1	Member	
<input type="checkbox"/>	 D	brittany.heigle@lyon.edu	CSC330 DATABASE THEORY AND APP... +1	Member	
<input type="checkbox"/>	 E	Dat O dat.o@lyon.edu	CSC330 DATABASE THEORY AND APP... +1	Member	
<input type="checkbox"/>	 F	emma.brandes@lyon.edu	CSC330 DATABASE THEORY AND APP... +1	Member	
<input type="checkbox"/>	 G	fortunatojose.herna@lyon.edu	CSC330 DATABASE THEORY AND APP... +2	Member	
<input type="checkbox"/>	 H	hakeem.yatim@lyon.edu	CSC330 DATABASE THEORY AND APP... +1	Member	
<input type="checkbox"/>	 I	hunter.perkins@lyon.edu	CSC330 DATABASE THEORY AND APP... +1	Member	
<input type="checkbox"/>	 J	Logan Richerson logan.richerson@lyon.edu	CSC330 DATABASE THEORY AND APP... +1	Member	
<input type="checkbox"/>	 K	Malcolm Howard malcolm.howard@lyon.edu	CSC330 DATABASE THEORY AND APP... +2	Member	
<input type="checkbox"/>	 L	Marlon Durand marlon.durand@lyon.edu	CSC330 DATABASE THEORY AND APP... +1	Member	
<input type="checkbox"/>	 M	moise.occulis@lyon.edu	CSC330 DATABASE THEORY AND APP... +1	Member	
<input type="checkbox"/>	 N	Spencer Rhoden spencer.rhoden@lyon.edu	CSC330 DATABASE THEORY AND APP... +1	Member	
<input type="checkbox"/>	 O	thienkim.ho@lyon.edu	CSC330 DATABASE THEORY AND APP... +1	Member	

Figure 6: DataCamp DB course member list

History of databases

- [YouTube video link](#)



Figure 7: Which of these founders has nothing to do with databases?

- DB are an old human interest (information is always gathered)

- DB development happened in close "combat" with companies and operating systems (market and a hardware aspect to it)
- Dominant language is SQL - in connection with relational DB
- Codd's 12 rules = everything you need to know about relational DB design (these rules scale extremely well)
- There are MANY different types of DBMS
- We'll be installing SQLite, too, for local SQL experimentation

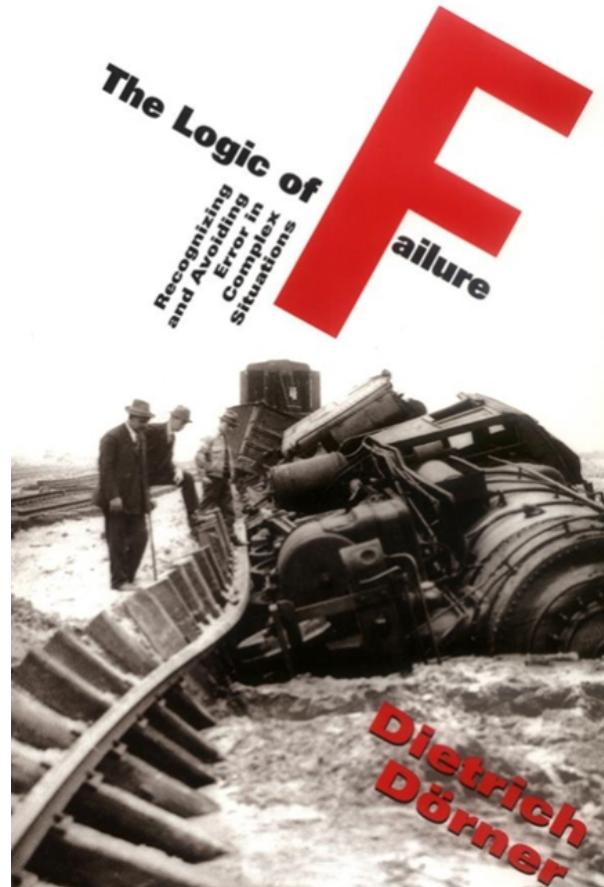
DBMS IoT example application (MooCall)

- Website for this [Irish app](#)



Figure 8: Two cows with MooCall sensors attached.

- Sensor applications are small and look trivial, but they're highly security relevant (Cernobyl disaster in USSR, 1986). See [The Logic of Failure \(Dorner, 1997\)](#).



- Big Data ("starts" modern data science) = 3Vs = Volume + Velocity + Variety [5V definitions add "value", "veracity"]
- IoT = "webservice"-enabled, cloud-networked, fast, big data applications
- MooCall DB networking:
 1. cow data are generated and pre-processed locally (e.g. cow ID, temperature, motion etc.) = "[edge computing](#)" example
 2. cow data are processed globally (in the cloud) to generate user signal ("this cow's calving!")
 3. Signal is transmitted to the user = farmer for potential action.
- DB system = DBMS + DB - the DBMS has a lot of fancy stuff on board: performance optimizer, shell, API
- BC (Before Codd) = file system-based data management, and AC (After Codd) = DB management system (DBMS)-based data management.

GNU Emacs installation

- Install vanilla¹ GNU Emacs or a modified Emacs (for statistics processing with R and other packages - [ESS](#))
- Vanilla GNU Emacs v27 Windows installation: you need the `emacs-27.1-x86_64-installer.exe` [from this page](#).
- MacOS: get the [modified version](#) if you like (easiest), or the binary using the command line terminal as shown [here](#), using the Homebrew package manager.

Index of /emacs/windows/emacs-27

	Name	Last modified	Size	Description
	 Parent Directory		-	
	 emacs-27-deps-mingw-w64-src.zip	2020-08-25 12:28	107M	
	 emacs-27-deps-mingw-w64-src.zip.sig	2020-08-25 12:28	833	
	 emacs-27.1-i686-installer.exe	2020-08-24 17:47	59M	
	 emacs-27.1-i686-installer.exe.sig	2020-08-24 17:47	833	
	 emacs-27.1-i686-no-deps.zip	2020-08-25 12:20	47M	
	 emacs-27.1-i686-no-deps.zip.sig	2020-08-25 12:20	833	
	 emacs-27.1-i686.zip	2020-08-25 12:22	102M	
	 emacs-27.1-i686.zip.sig	2020-08-25 12:22	833	
	 emacs-27.1-x86_64-installer.exe	2020-08-25 12:23	59M	
	 emacs-27.1-x86_64-installer.exe.sig	2020-08-25 12:23	833	
	 emacs-27.1-x86_64-no-deps.zip	2020-08-25 12:24	47M	
	 emacs-27.1-x86_64-no-deps.zip.sig	2020-08-25 12:24	833	
	 emacs-27.1-x86_64.zip	2020-08-25 12:26	103M	
	 emacs-27.1-x86_64.zip.sig	2020-08-25 12:26	833	
	 emacs-27.2-i686-installer.exe	2021-03-31 12:45	60M	
	 emacs-27.2-i686-installer.exe.sig	2021-03-31 12:45	833	
	 emacs-27.2-i686-no-deps.zip	2021-03-31 12:45	47M	
	 emacs-27.2-i686-no-deps.zip.sig	2021-03-31 12:45	833	
	 emacs-27.2-i686.zip	2021-03-31 12:45	102M	
	 emacs-27.2-i686.zip.sig	2021-03-31 12:45	833	
	 emacs-27.2-x86_64-installer.exe	2021-03-31 12:45	60M	
	 emacs-27.2-x86_64-installer.exe.sig	2021-03-31 12:45	833	
	 emacs-27.2-x86_64-no-deps.zip	2021-03-31 12:46	47M	
	 emacs-27.2-x86_64-no-deps.zip.sig	2021-03-31 12:46	833	
	 emacs-27.2-x86_64.zip	2021-03-31 12:46	103M	
	 emacs-27.2-x86_64.zip.sig	2021-03-31 12:46	833	

Figure 10: GNU Emacs v27 download online repo

- Demo: keyboard macros. I showed how to define a keyboard macro (`c-x ([key sequence] c-x)`) and how to apply it repeatedly (`c-u [times] c-x e`). [Here's the documentation](#).

DB elements, GNU Emacs - w2s3 (01/18/22)

Quiz 1

THE QUIZ IS ON ... SCHOOLOGY 1 PM - 1.15 PM

Feedback/discussion

FOLLOWED by brief FEEDBACK:

- We'll do one of these per week (I hope)
- Any content questions?
- Too much time? Too little?
- Questions too hard? Too easy?
- A subset of these questions will become the final exam
- After playing the quiz in class you can play it unlimited times
- Quizzes are now **ungraded** (final exam is now 30% of final grade)

Review: file vs database approach

DICT	
TABLE_NAME	COMMENTS
USER_RESOURCE_LIMITS	Display Resource limit to the User
USER_PASSWORD_LIMITS	Displays password limits to the User
USER_CATALOG	All tables, views,synonyms, Sequences owned by the users
ALL_CATALOG	All tables, views,synonyms, Sequences accessible to the users
USER_CLUSTURES	Description of user's own Clusters
ALL_CLUSTERS	Description of clusters accessible to other users
DBA_TABLES	Description of all relational tables in the database
DBA_ALL_TABLES	Description of all object and relational tables in the database
USER_TABLES	Description of the user's own relational tables
ALL_TABLES	Description of relational tables accessible to the user
ALL_INDEXES	Descriptions of indexes on tables accessible to the user
DICTIONARY	Synonym for DICTIONARY
COLUMN_PRIVILEGES	Grants on columns for which the user is the grantor, grantee, owner

Figure 11: database dictionary example (tutorialcup.com)

Captain's Log²

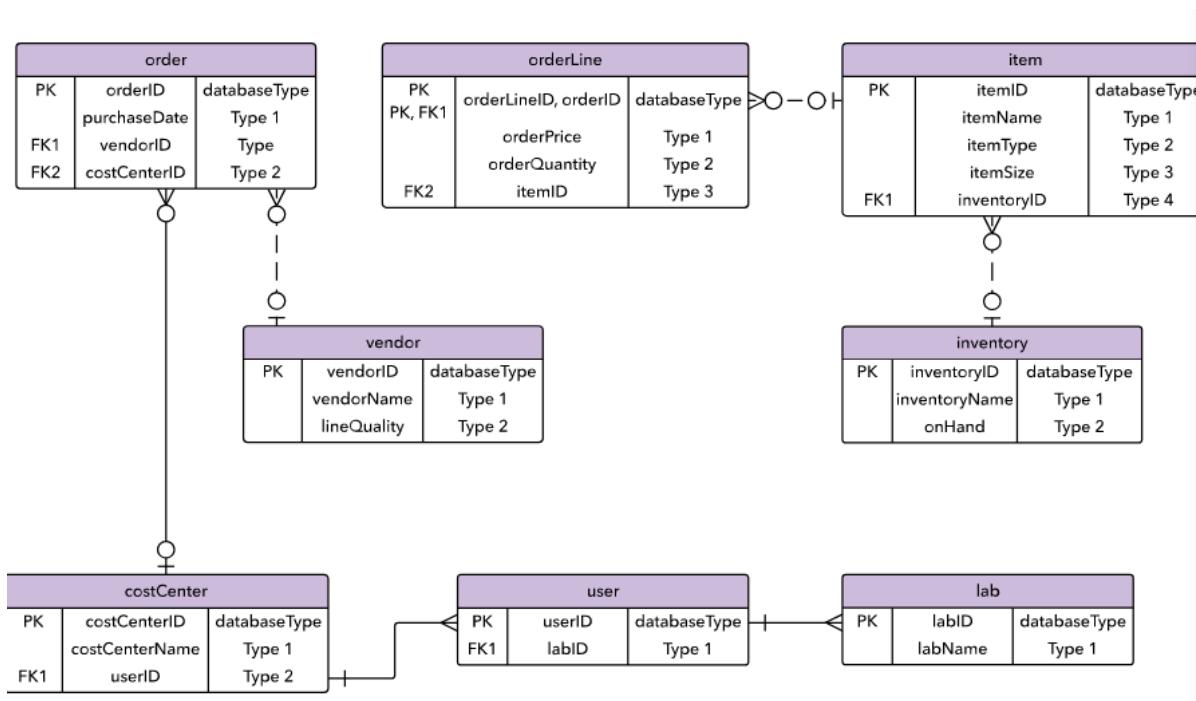


Figure 12: Entity Relationship Diagram (Source: Lucidchart).

- strong vs. soft/loose coupling is an important design issue (it relates directly to the resilience of the designed system).
- **meta data** are data about data, used to control, manage other data, and processes
- Example: [Exif \(Exchangeable Image File\)](#) data are an example of common meta data associated with image files stored alongside digital pictures.
- SQL (Structured Query Language): dominant language for structured, table-based (aka relational) databases
 - DQL = Data Query Language (e.g. selecting data)
 - DDL = Data Definition Language (e.g. creating tables)
 - DML = Data Manipulation Language (e.g. input/output of data)
 - DCL = Data Control Language (to alter meta data, e.g. rights)
- Learning anything is best as a "variation on a theme" (as in music)
- MooCall: what do we want to store (cow temperature F, in C)
- State-based view of computing (von Neumann architecture) = dominant paradigm (automata, Turing machines...)

GNU Emacs tour

- Ctrl-h Ctrl-a RET : Startup screen
- Emacs written in C and Lisp (Emacs Lisp)
- Emacs is an IDE - we'll use it for SQL, SQLite, and bash
- Emacs contains a bunch of apps (e.g. file explorer)
- Try the Emacs onboard tutorial (CTRL-h t)
- What Emacs can do:
 - Extension and full customization (with Emacs Lisp)
 - Writing in many different human/programming/markup languages (with major and minor modes)
 - IDE work (compile, run, test programs) - gdb integration
 - Compare and highlight file differences (with ediff)
 - Manage files (with dired)
 - Read mail, news, RSS feeds (gnus)
 - You can use it as an IRC reader (#batesville@irc.freenode.net)
 - Play games ([examples](#))

Whenever you decide to start using Emacs, you should take the Emacs tutorial. It's an interactive hands-on which will familiarize you with many things, including:

- Starting and exiting Emacs
- Basic text movement and editing commands
- Opening and saving files
- Emacs concepts: windows, frames, files, and buffers
- Invoking commands with keybindings and with M-x
- To run the tutorial, start Emacs and type C-h t, that is, Ctrl-h followed by t.

SQLite installation - w2s4 (01/20/22)

Captain's Log

- GNU Emacs reference card ("cheat sheet") [on GitHub \(PDF\)](#)
- We looked at different reasons to use the Emacs editor/IDE
- For installation of the software we need (Emacs, SQLite), which is already done on the PCs in the computer lab, check [install.org](#) in GitHub

- To get better at moving through Emacs buffers, manage files etc., complete the GNU Emacs onboard tutorial (open it in Emacs with `C-h t`)
- We'll go through a more systematic training session (including Org-mode) next week.

Cloud computing intro - w3s5 (01/25/22)

Cloud computing - "the old curmudgeon's view"

- Spending on cloud services in 2020: \$bn 266 - projected to \$bn 308 in 2021 (DataCamp). However ([IDC, 2022](#))

For the full year 2021, IDC forecasts cloud infrastructure spending to grow 8.3% compared to 2020 to \$71.8 billion, while non-cloud infrastructure is expected to grow 1.9% to \$58.4 billion after two years of declines. Shared cloud infrastructure is expected to grow by 7.2% year over year to \$49.7 billion for the full year. Spending on dedicated cloud infrastructure is expected to grow 10.7% to \$22.2 billion for the full year.

Figure 13: Cloud service spending data (IDC, 2022)

- What's the GDP of [Finland](#) and [Vietnam](#)?
- On-premise vs cloud discussion - major business issue - why?
- Main messages: cloud is huge, good for access, scaling is an issue
- Amazon Web Services is highlighted (why?) What about other providers? How do they differ? What kind of market is this?
- What do you think of the AWS products page from a logical point of view?

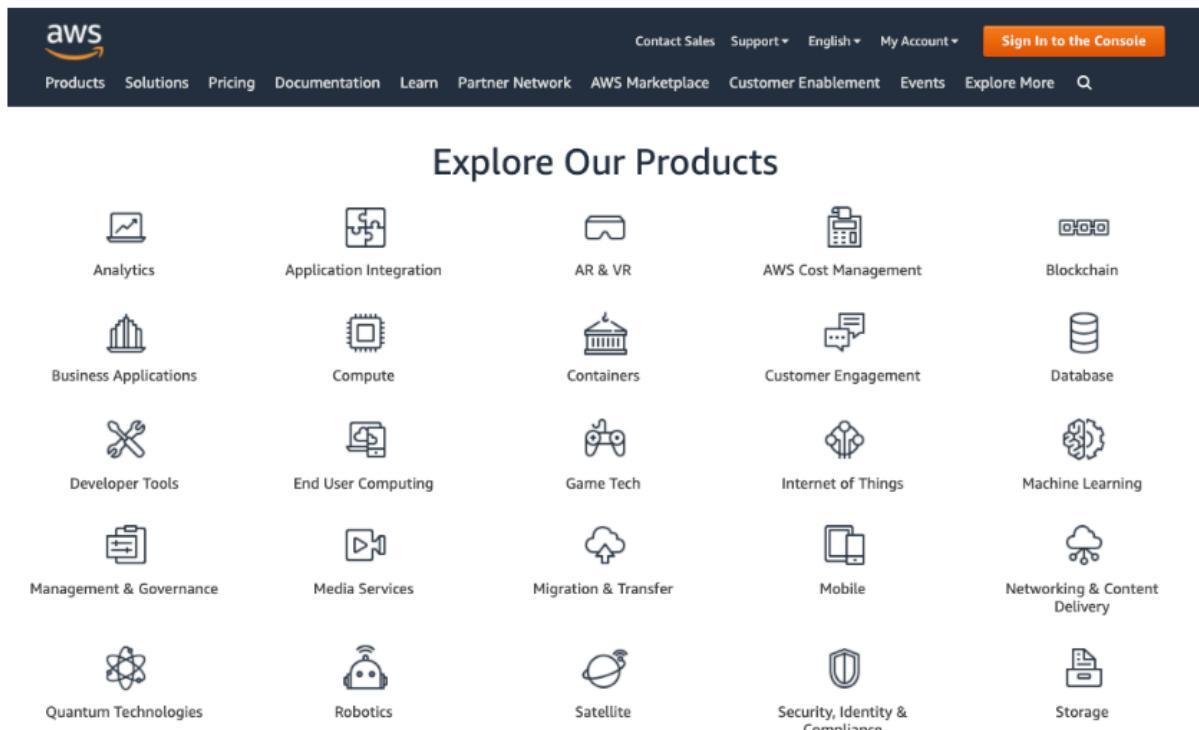


Figure 14: aws products and services (DataCamp, 2020).

- Virtualization = your OS over the internet
- Vertical (server power) vs. horizontal scaling (server number)
- Data centers - cp. "["Inside Google's Data Centers"](#) (H5P/YouTube)
- How are Lyon's data served - what about the safekeeping?

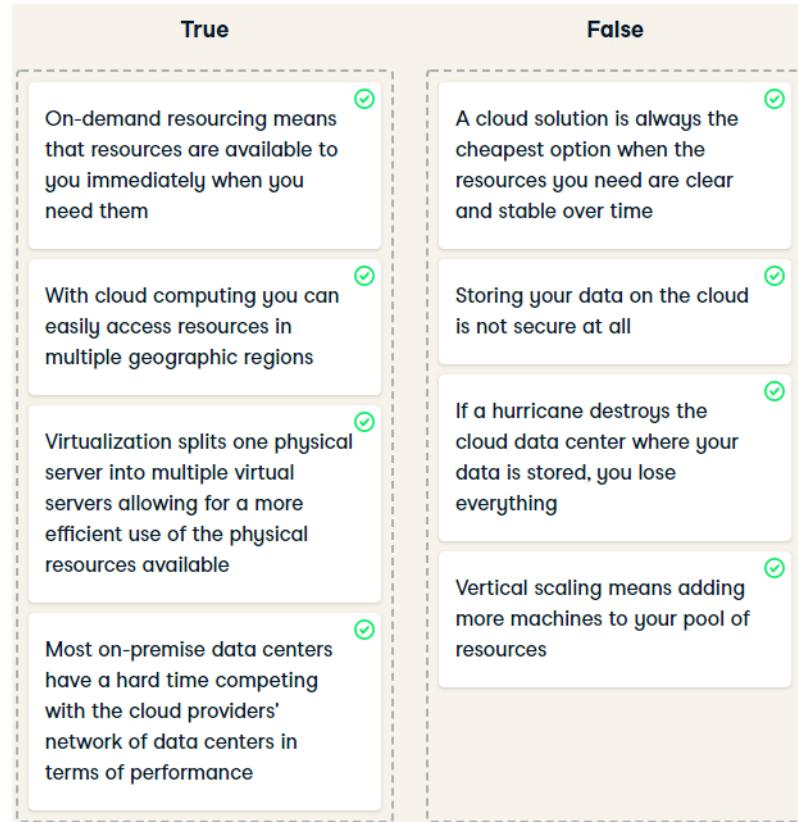


Figure 15: On-demand cloud vs. on-premise data centers

- IaaS/PaaS/SaaS: Socialist propaganda or brave new world of shared services? What're some key conditions for this to be beautiful?

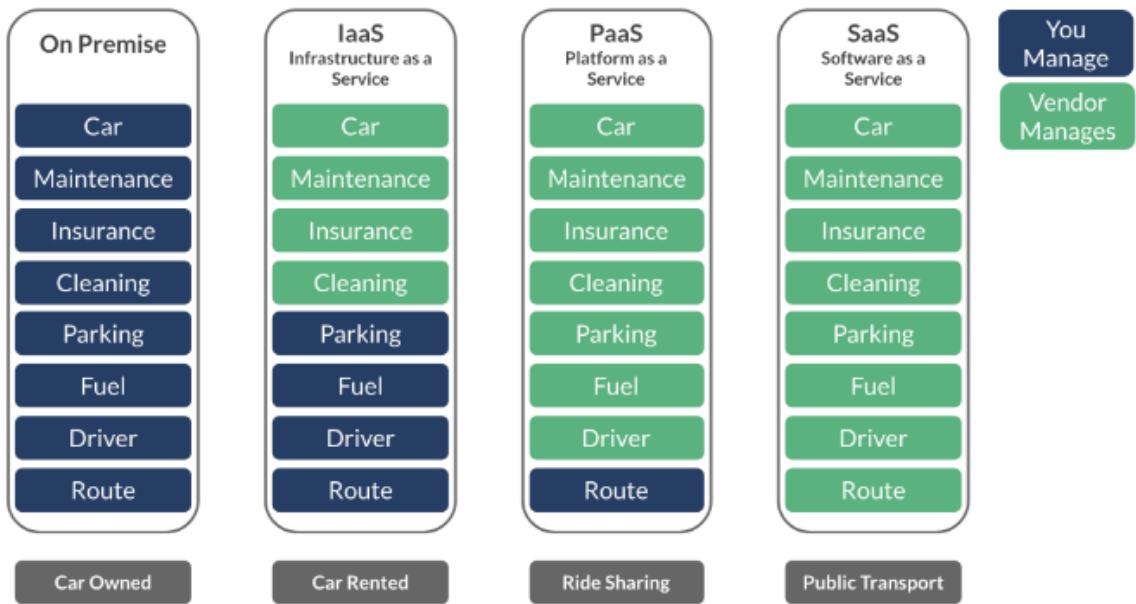


Figure 16: Cloud service models - car analogy (DataCamp, 2020)

- IT services

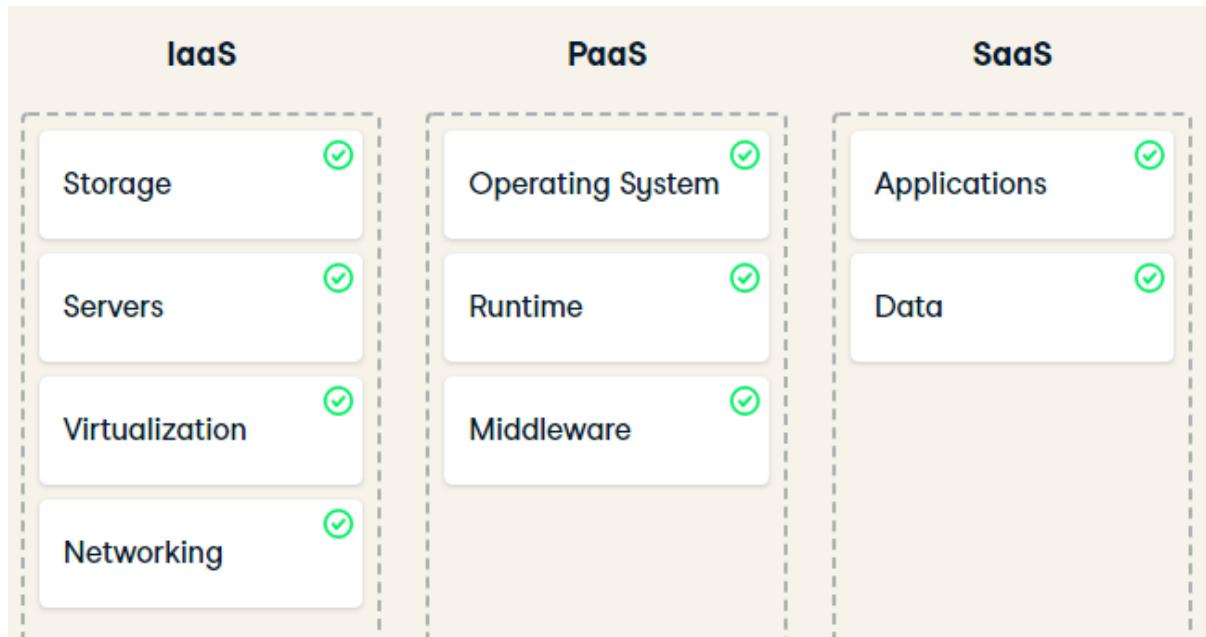


Figure 17: IaaS vs. PaaS vs. SaaS (DataCamp, 2020)

- Use cases (UML anyone?)

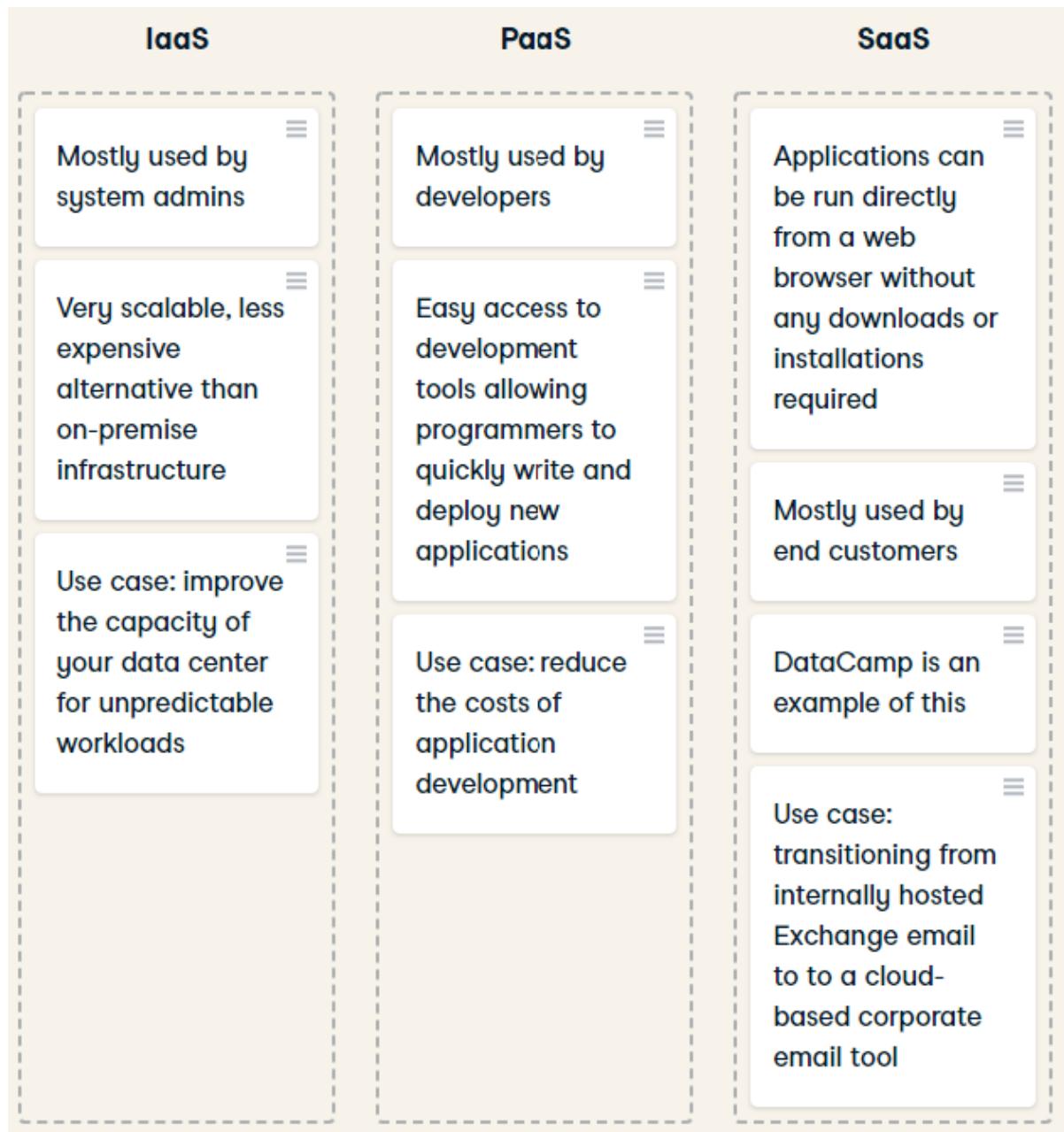


Figure 18: IaaS vs. PaaS vs. SaaS (DataCamp, 2020)

- The cloud pyramid - "More complexity means less control"



Figure 19: Control vs. Abstraction (DataCamp, 2020)

Captain's Log Star Date 99667.82

- Use case diagrams are part of the Unified Modeling Language (UML). Here is a good, short, free [online video tutorial \(Lucidchart, 2018\)](#). UML is a visual language to describe information systems.
- We discussed a bank (more specifically a retail, or consumer bank) as an example. It is not a good use case for cloud computing, because the data are confidential, structured, don't change much, and are small. Banks more often use on-premise relational database management systems (RDBMS) and SQL, than cloud-based systems.
- When we ask (about a table, or a diagram, or a graph, or any scheme), "is this logical?", we mean "is this well ordered, or ordered at all, fit for human understanding." There is a whole method, the so-called "Minto Pyramid Principle" built around this concept of ordering content so that it is fit for human understanding. Here is a short [online video tutorial on Minto \(Harrison Metal, 2019\)](#).
- "Shell inside a shell": when we open the Windows terminal, or Command prompt (cmd .exe), and then start a program like sqlite3, we operate in a shell inside another shell.
- REPL = Read-Eval-Print-Loop - cloud-based application to learn programming: [replit.com](#).

Cloud deployment - w4s6 (02/01/22)

DataCamp assignment - cloud deployment

- Deployment models: private, public, hybrid, multicloud, community
- Negative personal example (hybrid): RStudio cloud
- Positive personal example (public): Colaboratory + GDrive, DataCamp Workspace
- Nice exercise: build a NextCloud server with Raspberry Pi
- EU GDPR - the true story: a major pain with unclear gains
- Is there a US equivalent of the General Data Privacy Regulation act of the EU?³
- "What is personal data?" is an interesting question - why?⁴
- How international is the Internet really? (What is its backbone, where are public data held, and who owns the infrastructure?)
- How do "cloud computing roles" relate to "database roles"?
- How can you improve your "cloud computing skills"? What are they? (Examples: [IBM@coursera.org](#), [EdX cloud computing courses](#), [Google](#))

- Overlap with data science: analysis (EDA), engineering (pipeline building and maintenance), and modeling (ML)
- Database apps: Tableau etc. all based on SQL = top skill
- Buzzword (2010-2020): "Digital transformation of the company"

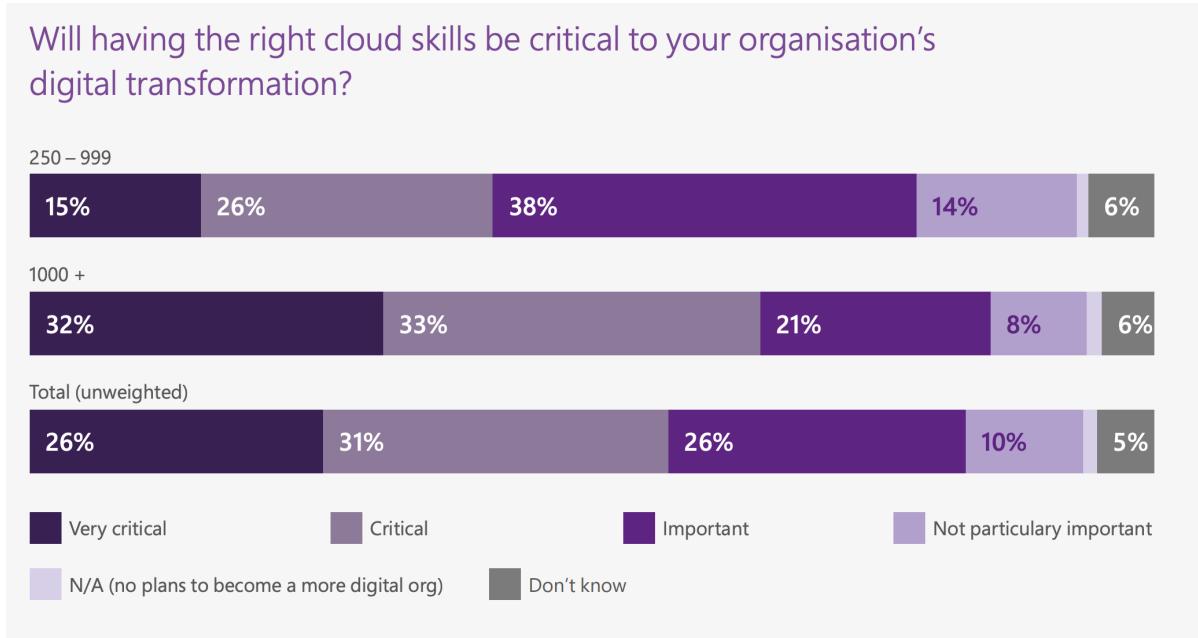


Figure 20: Microsoft Skills Report (2017)

Captain's Log Stardate 99687.04

- If you don't have the Linux program touch on Windows, you can use this command to create an empty file in the current directory:

```
$ fsutil file createNew test.db 0
```

- An even simpler way is by opening the RDBM program and create a DB at the same time with the command sqlite3 test.db at the prompt.
- [NextCloudPi](#) contains documentation to use the cloud computing software NextCloud on the Raspberry Pi. Somewhere in there you'll find what it takes to build a backup server with NextCloud (which was one of my projects).
- [World-Wide Web vs. Internet \(BBC, 2019\)](#):

"The World Wide Web Is Not The Internet!"

NEXT SQLite introduction - w5s7 (02/08/22)

DataCamp assignment - Cloud providers and case studies



Figure 21: The benefits of French culture (Source: ila-france.com)

Disclaimer (limitations / bias statement)

- "Customers churn"?
- Commercial examples (no scientific objectivity)
- Case studies with a strong French accent (your reaction?)

Amazon Web Services	<input checked="" type="checkbox"/>
Microsoft Azure	<input checked="" type="checkbox"/>
Google Cloud	<input checked="" type="checkbox"/>
Alibaba Cloud	<input checked="" type="checkbox"/>
IBM Cloud	<input checked="" type="checkbox"/>
Oracle Cloud	<input checked="" type="checkbox"/>

Figure 22: Global cloud service providers (DataCamp, 2019)

AWS - Amazon Web Services

- Q4/2019 - pre-pandemic (changes in 2021?⁵)

WHAT	2019	2021
market share	32.4%	51.1%
revenue	\$9.8bn	<u>\$17.8bn</u>
annual revenue growth	33.2%	<u>40%</u>

- First mover advantage
- Are there jobs with AWS in Arkansas?
- Does [case example Nerdwallet](#) (FinTech) still exist?

Microsoft Azure

- Q4/2019 - pre-pandemic (changes in 2021?⁵)

WHAT	2019	2021
market share	17.6%	31.9%
revenue	\$5.3bn	\$60bn
annual revenue growth	62.3%	<u>51%</u>

- Azure integrates with Microsoft products
- Are there jobs with Azure in Arkansas?
- Case example: [Ottawa Hospital](#) (disaster recovery)

Google Cloud Services

- Q4/2019 - pre-pandemic (changes in 2021?⁵)

WHAT	2019	2021
market share	6%	9.1%
revenue	\$1.8bn	\$65.12bn
annual revenue growth	67%	41%

- Are there jobs with Google Cloud in Arkansas?
- Lush case example (cosmetics retailer): compensate traffic fluctuations on their e-commerce platform

Other actors

- Alibaba Cloud (China) - 5.4% (2021: 8.9%) - AI focus
- IBM - 1.8% - SPSS (stats) + Watson (health care)
- Oracle Cloud services (owns open source MySQL, VirtualBox)

How does one pick a cloud provider?

- Current infrastructure costs
- Cost of running a data center
- Cost of applications
- Cost of hiring cloud specialists
- Benefit for company and customers
- Risks (regulation, security, disaster, lock-in)

What's the right order (from most important to least important)?

Future of the cloud

- Data trends (cp. [2022 Data trends by DataCamp](#))
- [Cryptocurrencies \(Smalley, 2021\)](#).
- Post-pandemic effects

After finishing a DataCamp course:

1. download the slides and review if necessary
2. download the certificate and add it to your resume

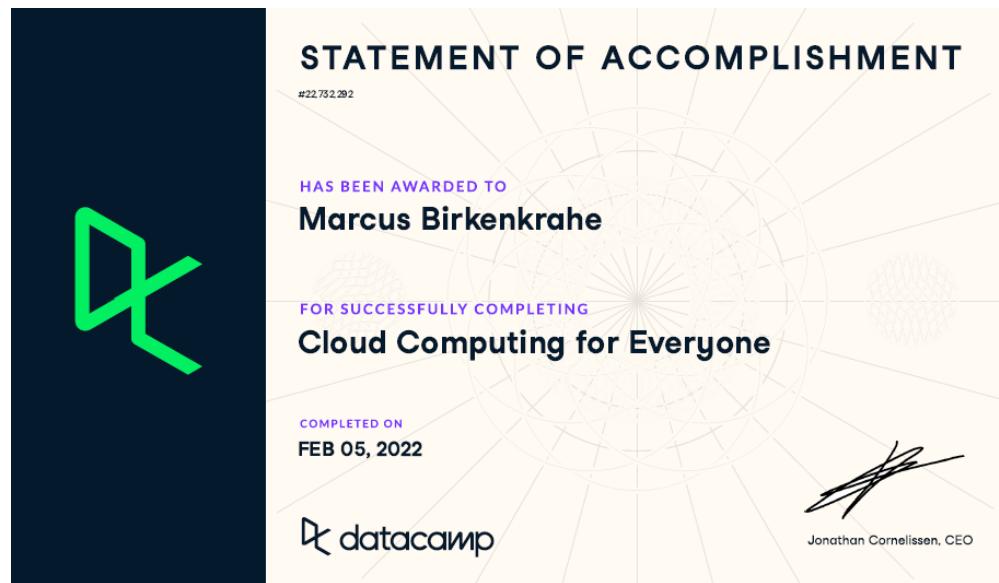


Figure 23: Statement of Accomplishment (DataCamp, 2022)

TODO Introduction to SQLite

TODO SQLite Practice

Glossary = the learning dictionary for your brain

- Why is it important to have a glossary?

TERM	MEANING
------	---------

TERM	MEANING
Meta data	Data about data, e.g. control information for a database
DDL	Data Definition Language
DML	Data Manipulation Language
DQL	Data Query Language
DCL	Data Control Language
Coupling	(design) Relates to the independence of parts of a system
gdb	GNU debugger, supports many languages
IaaS	Infrastructure as a Service (roads)
PaaS	Platform as a Service (roads + shops)
SaaS	Software as a Service (roads + shops + products)
FaaS	Function as a Service
Scaling	Horizontal or vertical - increase computing performance
On-premise	Application + data reside in the company or organization
Cloud	Application + data reside in the network
Virtualization	Split the server action up across different locations
GDPR	General Data Privacy Regulation act of the European Union
Internet	Global digital infrastructure (router, network)
Web	Global collection of digital content (browser, wiki)

References

- BBC (11 Mar 2019). World wide web vs. internet - what's the difference? [video]. [URL: www.bbc.co.uk](http://www.bbc.co.uk).
- Birkenkrahe (Jan 11, 2022). Interactive shell vs. interactive notebook (literate programming demo). [URL: youtu.be/8HJGz3IYoHI](https://youtu.be/8HJGz3IYoHI).
- bnewall1 (Apr 25, 2010). Star Trek: Captain's Log - 11/30/1994 - 2/7 [video]. [URL: youtu.be/T2bSMLEQX1o](https://youtu.be/T2bSMLEQX1o).
- Smalley (22 Oct 2021). Cryptocurrency is changing the Data Center Market [blog]. [URL: datacenters.com](http://datacenters.com).
- Harrison Metal (2019). Thank You, Barbara Minto [video]. [URL: vimeo.com/305393045](https://vimeo.com/305393045).
- Lucidchart (2022). What is a Database Model [website]. [URL: www.lucidchart.com](http://www.lucidchart.com).
- Lucidchart (Feb 7, 2018). UML Use Case Diagram Tutorial [video]. [URL: youtu.be/zid-MVo7M-E](https://youtu.be/zid-MVo7M-E).
- Moocall (Nov 27, 2020). Moocall Calving Sensor [video]. [URL: youtu.be/718uGYbUmao](https://youtu.be/718uGYbUmao).
- NextCloudPi Documentation (2019). Welcome [website]. [URL: docs.nextcloudpi.com](https://docs.nextcloudpi.com).
- Nystrom (2021). Crafting Interpreters. Genever Benning. URL: <https://craftinginterpreters.com/>
- Russell/Norvig (2021). Artificial Intelligence - a Modern Approach (AIMA). Pearson. URL: aima.cs.berkeley.edu.
- Spajic (Jan 29, 2018). Understanding Git - Data Model. [URL: medium.com/hackernoon](https://medium.com/hackernoon).
- Valencia (Feb 28, 2017). The Lisp approach to AI (Part 1). [URL: medium.com/ai-society](https://medium.com/ai-society).

Footnotes:

¹ The term "vanilla" refers to the fact that this is the uncustomized, original version of Emacs. For large, old open source projects, this is a bit of a mystery, though, since so many versions exist that it may be difficult to identify "the original".

² ["Logic, logic...I'm sick to death of logic!"](#) (Star Trek: Captain's Log)

³ There is no federal data privacy law like GDPR in the United States. There are some national laws that have been put in place to regulate the use of data in certain industries, e.g. The 1974 U.S. Privacy Act (rights and restrictions regarding data held by US government agencies), or the 2018 California Consumer Privacy Act (rights and protection for CA residents).

⁴ Privacy is a political term. Private information can also be an entry to phishing or other network-based activities compromising the security of your PC.

⁵ [Statista](#) numbers (no source given) for 2021.

Author: Marcus Birkenkrahe

Created: 2022-02-08 Tue 20:38

[Validate](#)