CS6218. Principles of Programming Languages & Software Engineering

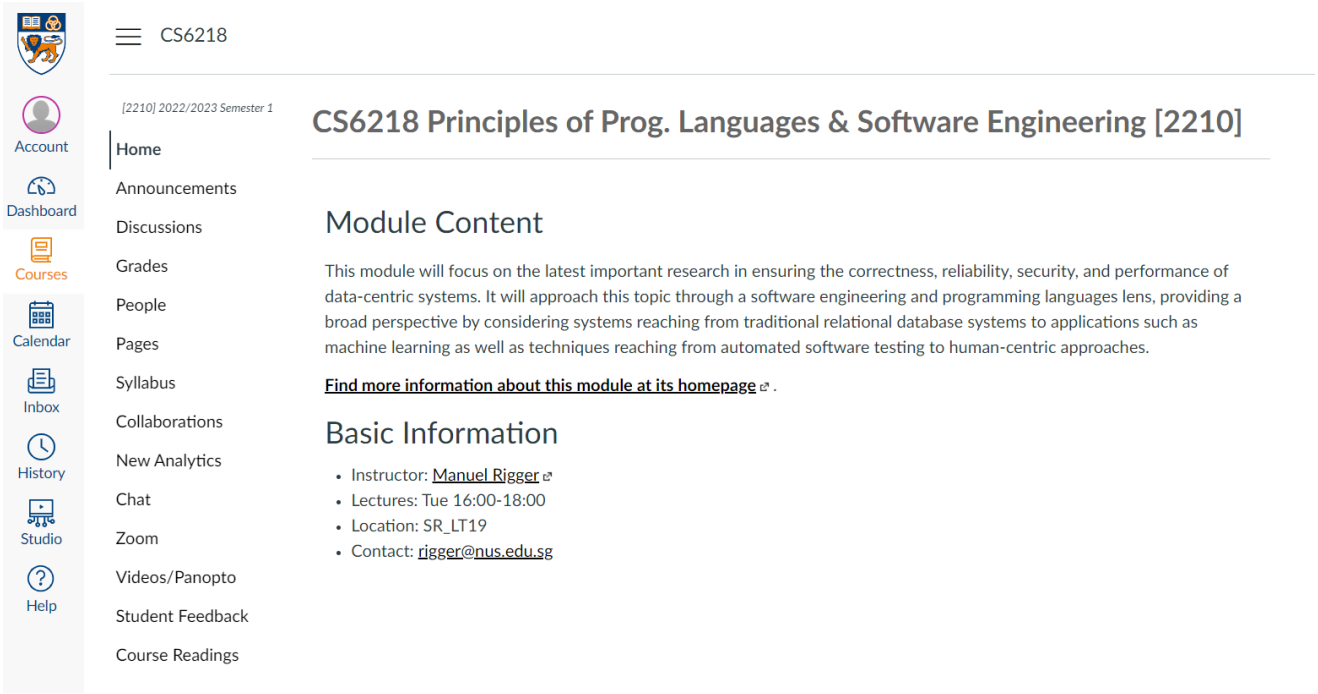
Week 1: Logistics & Motivation

Logistics & Overview

Logistics

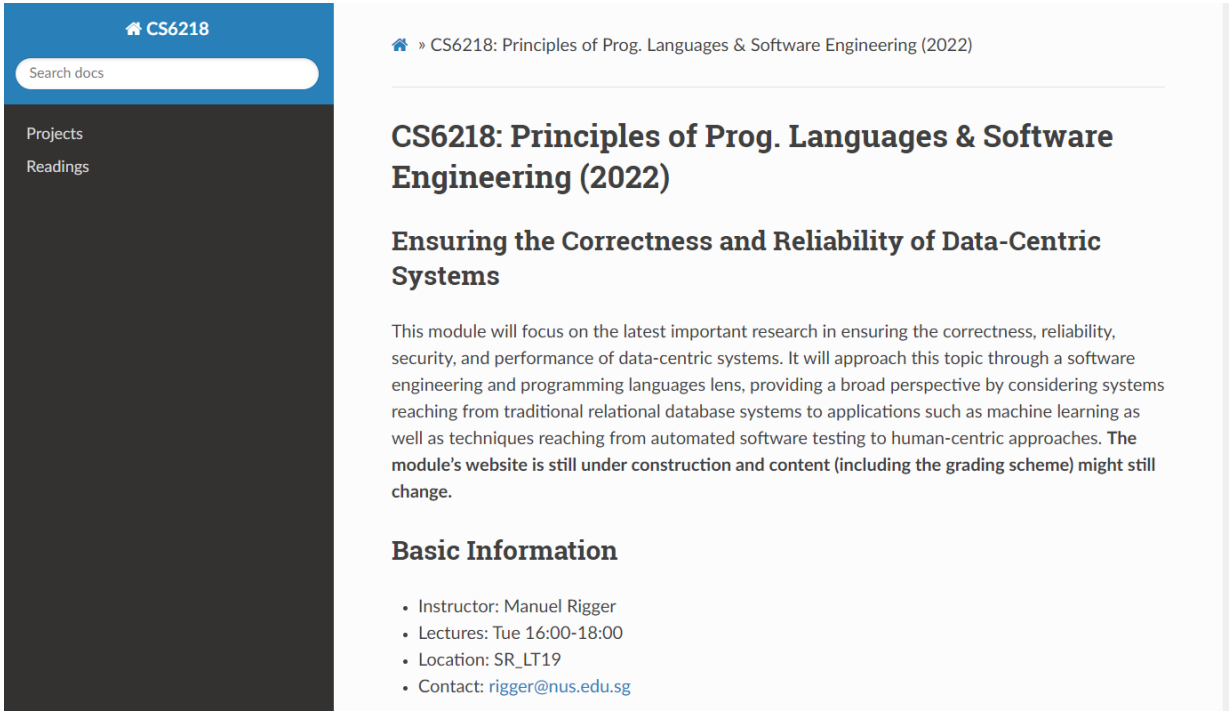
- Instructor: Manuel Rigger (李曼努)
- Lectures: Tue 16:00-18:00
- Location: SR_LT19
- Contact: rigger@nus.edu.sg

Module Information



https://canvas.nus.edu.sg/courses/25019

Module Information



https://manuelrigger.at/teaching/CS6218/

Module Contents

This module will focus on the latest important research in ensuring the correctness, reliability, security, and performance of data-centric systems. It will approach this topic through a software engineering and programming languages lens, providing a broad perspective by considering systems reaching from traditional relational database systems to applications such as machine learning as well as techniques reaching from automated software testing to human-centric approaches.

SE/PL Perspective

- Techniques: automated testing, debugging, verification, …
- Publication venues: ICSE, ESEC/FSE, PLDI, POPL, ASPLOS, OOPSLA, …
  - But also other venues in which relevant works were published!
- Approaches: white-box and grey-box

About Me

- Enthusiastic about this topic!
  - Lab's focus (https://nus-test.github.io/)
- Designed and developed SQLancer to find logic bugs in database systems
  - Found more than 500 bugs in widely used database systems
- PC-chair of DBTest '22, Co-organizer of Dagstuhl seminars on "Ensuring the Reliability and Robustness of Database Management Systems"

https://github.com/sqlancer/sqlancer

Module Goals: Primary Goals

- Learn about general techniques such as differential testing, metamorphic testing, debugging techniques, programming languages & type systems, etc.
- Learn how these and other techniques can be applied in the context of data-centric systems

Module Goals: Secondary Goals

- Build skills on how to read and judge scientific projects/papers
- Develop a taste on research
- Practice giving talks and giving feedback

Grading Scheme

- 40% project
- 40% presentation
- 20% attendance and participation

Grading Scheme

- 40% project
- 40% presentation
- 20% attendance and participation

Project

- Alone or groups of two (document shared effort via Git)
- Goal: gain hands-on experience
- Research-focused or implementation-focused
  - Graded accordingly
- Deliverable
  - Initial project proposal
  - Final report & project overview presentation (recording or in-person)

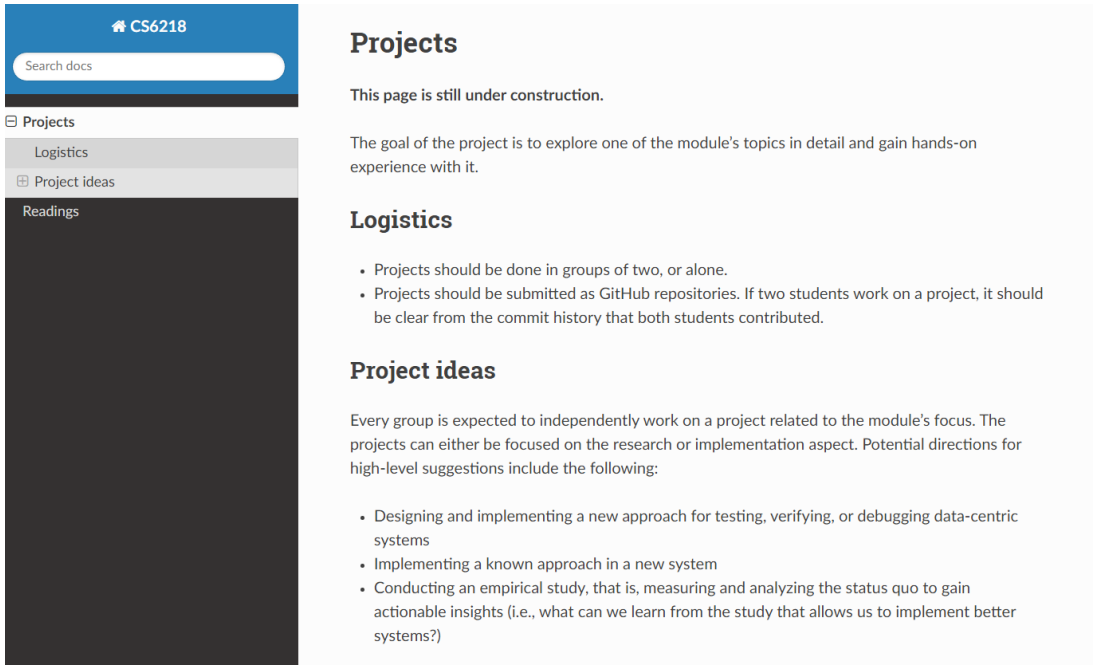
Reports

- acmart with sigconf (Latex:`\documentclass[sigconf]{acmart}`)
- https://www.acm.org/publications/proceedings-template
- Page Limits
  - Initial project proposal: ~1 page
  - Final report: 4 pages

Initial Project Proposal

- Goal: obtain some feedback from me
- Questions that should be answered
  - What is the problem being tackled?
  - Why is the problem important?
  - Is the project research-focused or implementation-focused?
  - What is the proposed solution?
  - What is the proposed timeline?
  - What is the expected outcome of the project?

Initial Project Proposal



https://manuelrigger.at/teaching/CS6218/projects.html

(Self) Plagiarism

- Fine to build on previously submitted projects (e.g., for another module)
  - But: highlight the proposed differences and enhancements in the report
- No paper presentation about a paper previously presented

Final Report

- Follow a typical paper structure, for example:
  - Abstract
  - Introduction
  - Approach
  - Implementation
  - Evaluation
  - (Related Work)
  - Conclusion

Project Presentation

- 15 minutes time limit
- Give an overview of the project
- If sufficient time, we'll have in-class presentations

Grading Scheme

- 40% project
- 40% presentation
- 20% attendance and participation

Presentation

- Everyone will present 1~~–2 papers~~
  - Number depends on the number of attendees
- Subsequent Q/A and presentation feedback

Presentations: Paper Selection

- https://nus-cs6218-2022.hotcrp.com/reviewprefs
  - System used for paper reviewing
  - We will use it only for bidding and assigning papers
- More papers than attendees to account for different interests

Presentations: Other Papers

- Inherently interdisciplinary topic with many directions and strands of research
- Suggest an alternative paper that you would like to present

Designing Presentations

- “How to Give a Great Research Talk”
- By Simon Peyton Jones

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

Designing Presentations

- “How to Design Talks”
- By Ranjit Jhala

https://www.youtube.com/watch?v=aFT79TmffPk

Designing Presentations

Andreas Zeller

Andreas Zeller is faculty at the CISPA Helmholtz Center for Information Security and professor for Software Engineering at Saarland University, both in Saarbrücken, Germany. His research on automated debugging, mining software archives, specification mining, and security testing has proven highly influential. Zeller is an ACM Fellow and holds an ACM SIGSOFT Outstanding Research Award.

office-zeller@cispa.de
+49 681 87083-1001
@AndreasZeller

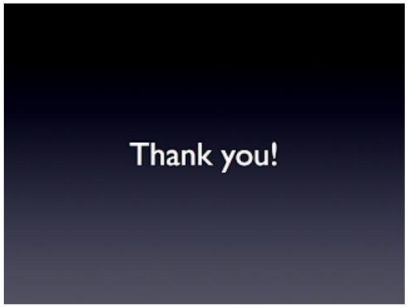
22 October 2013

Summarizing your presentation with miniature slides

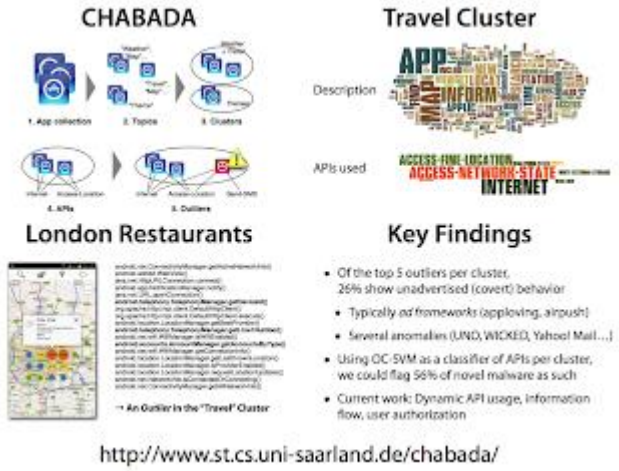
by Andreas Zeller

The slide that drives me nuts

As part of my job, I listen to talks a lot. There's good talks, and there's bad talks. But nothing can drive me as crazy as making a slide like *this* your final slide:



What's wrong with showing "Thank you!" at the end? Or "Questions?"? Or both? The problem is that most talks are followed by a discussion. With a slide like this showing up at the end, there's three problems:



https://andreas-zeller.info/2013/10/22/summarizing-your-presentation-with.html

Designing Presentations

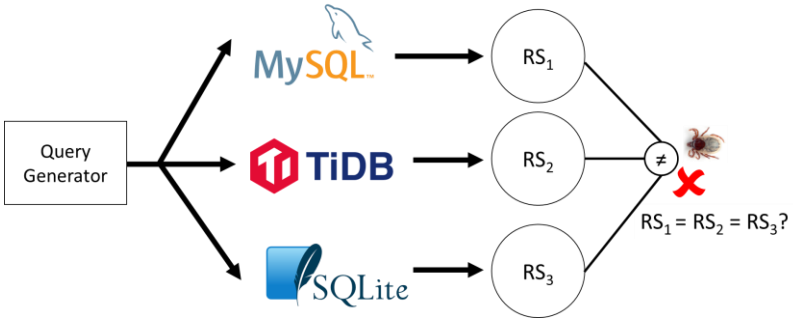
How could we visualize this?

- Use visual elements rather than text
  - Vector graphics whenever possible
- Use own examples and figures to demonstrate a deep understanding
- Including a motivating/running example can help with organizing the presentation
- Making good slides and practicing talks needs lots of time

Designing Presentations: Visual Elements

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque porttitor, lorem quis cursus gravida, sapien augue venenatis mi, sed semper felis velit a odio. Aenean imperdiet lacus non ipsum commodo feugiat. Nunc sodales in neque id laoreet. Sed sagittis, augue eget mollis suscipit, nisi tellus ullamcorper erat, et tincidunt dui turpis eu dolor. Cras quam tortor, mattis vel ullamcorper in, tincidunt convallis urna. Sed interdum, risus at ultricies venenatis, metus felis viverra magna, at viverra nulla nulla at nisi. Ut semper molestie varius. Nunc sed bibendum eros. Curabitur ultricies, massa et imperdiet fringilla, ante tortor lacinia massa, non euismod sapien orci nec dui. Phasellus nec dignissim leo. Vivamus lacinia tellus et lorem malesuada fermentum. Sed justo tellus, tempus in porttitor ut, laoreet ac mauris. Cras scelerisque libero nec risus ullamcorper, et convallis ex eleifend. Nunc dictum velit venenatis lacus auctor rhoncus. Aliquam sodales cursus metus, in rutrum nisi laoreet et. Donec pulvinar rutrum nulla, sit amet aliquet magna laoreet sed. Vestibulum mattis, odio pulvinar blandit tincidunt, magna nisi pharetra justo, pretium consequat libero eros nec dolor. Maecenas eget nibh sem. Sed nunc leo, tincidunt sit amet condimentum ultricies, porttitor a libero. Etiam placerat blandit odio, vel viverra libero rhoncus id. Aliquam tincidunt finibus est in ullamcorper. Morbi rutrum, ante ut vestibulum elementum, tellus mi laoreet tortor, non interdum ipsum nibh sed risus. Pellentesque at purus nec orci elementum sollicitudin. Cras at metus vel augue elementum tincidunt ut volutpat ipsum. Nulla facilisi. Cras suscipit convallis ante vitae congue. Donec ultricies eros non nunc vehicula, vitae blandit velit ultrices. Mauris est nibh, pretium sed posuere sagittis, tristique vitae lectus. Integer ullamcorper sed massa et maximus. Duis ornare nec libero quis congue. Etiam et mi mollis, blandit nunc et, imperdiet sem. Vestibulum lacus sapien, sollicitudin eget semper quis, maximus at lacus. Curabitur eget ligula vitae massa porttitor eleifend. Sed sed dui vitae lacus vehicula venenatis. Nulla tempus hendrerit eros, consequat elementum felis. Morbi aliquet sem ac neque dapibus, nec lobortis dolor lobortis. Etiam ut est nec magna mollis rutrum. Ut aliquam interdum tristique. Pellentesque finibus ipsum et mi auctor, quis mollis dui porta. Etiam vitae nibh elit. Duis vulputate arcu elit, eget ornare sapien tristique sed. Sed imperdiet odio vitae mollis fringilla. Sed aliquet fringilla nulla vitae dapibus. Suspendisse consequat finibus lacus, sed fringilla leo molestie id. Integer non tristique odio, in pharetra nulla. Nunc varius urna a dolor vestibulum, eget fringilla lorem accumsan. Praesent non eros malesuada libero tristique rutrum. Pellentesque blandit lectus odio, vel dignissim nisi ultricies sit amet. Cras sit amet sapien nibh. Curabitur ut mi vel tellus auctor facilisis ac at odio. Vivamus vulputate lorem vel ligula cursus elementum. Sed venenatis magna at mattis elementum. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Aliquam id lectus imperdiet, ultrices est viverra, ornare nulla. Vestibulum turpis nisi, dictum id lobortis eget, ornare ut justo. Maecenas ut tristique orci, ac viverra elit. Maecenas sit amet justo pellentesque lorem ultrices porttitor. Sed sed eros quis nisl tempor vehicula vel sed dui. Nulla bibendum consectetur est, sed dapibus massa malesuada non. Quisque euismod ex erat, nec tincidunt odio semper ut. Ut pretium magna eget nisl semper condimentum. Morbi sit amet suscipit lectus, non ullamcorper neque. Sed pharetra odio quam, et faucibus dui condimentum at. Mauris mi tortor, placerat posuere magna in, faucibus sagittis tellus. Integer mollis erat sed tortor rutrum congue. In rhoncus ultricies porta. Sed luctus ipsum vel ultricies venenatis. Donec pharetra eu massa nec rhoncus. Nunc cursus lorem nec velit faucibus viverra. Etiam sollicitudin, neque sed congue viverra, mauris dui laoreet massa, nec vestibulum orci diam non arcu. Morbi eu vestibulum mi, posuere sollicitudin sapien. Nulla elementum id turpis in dapibus. Duis fermentum orci ac sollicitudin sagittis. Vivamus iaculis tempus ex, sed aliquet lectus. Praesent ultricies lectus id leo laoreet, non fringilla quam convallis. Curabitur accumsan posuere feugiat. In nisi dui, hendrerit ac tristique ac, vehicula non quam. Curabitur id fringilla ante. Aliquam lobortis tortor ipsum. Nunc luctus, dui at molestie fringilla, nisl leo semper orci, quis venenatis justo nisl ut risus. Ut id arcu velit. Vestibulum ultricies sodales est, eu sodales augue hendrerit vitae. Pellentesque nec magna a nisi tempor dapibus non luctus arcu.

"Wall of text"



(Simple) figures

Designing Presentations: Vector Graphics

Pixelated images

Vector graphics

Designing Presentations

- Use visual elements rather than text
  - Vector graphics whenever possible
- Use own examples and figures to demonstrate a deep understanding
- Including a motivating/running example can help with organizing the presentation
- Making good slides and practicing talks needs lots of time

Giving Feedback

- Giving constructive feedback is difficult
- Frameworks can help
  - What/why
  - Feedforward
  - Feedback sandwich

Giving Feedback: What/why

- Explain what the presenter did and why it was effective (or not)
- Example 1: You used many visuals, which made the presentation lively and easy to understand.
- Example 2: You were speaking fast, which made it difficult to follow the contents.
- Issue: giving direct negative feedback can be difficult to give and digest

Giving Feedback: Feedforward

- Focus on the future rather than past
  - More positive
- Example: For the next talk, it would be helpful to speak slower and add some breaks.

Giving Feedback: Feedback Sandwich

- Start with some positive feedback
- Provide constructive criticism
- End on a positive note
- Example: I overall liked the talk. It was clear to the audience that you put much work in designing the examples and figures. Despite this, it was difficult to follow the talk since you talked very quickly. I think by trying to add some breaks, you could make your future talks even better.

Grading Scheme

- 40% project
- 40% presentation
- 20% attendance and participation

Attendance

- One unexcused no-show allowed (i.e., no need to inform me)
- More for valid reasons (e.g., paper presentation at a conference)

Timeline

- Proposing additional papers: August 17
- Paper bidding: August 18
- Team composition: September 15
  - One day before the project proposal submission to set up Canvas
- Project proposal: September 16
  - Finish before recess!
- Final report & recording: TBD

Timeline

- week 02 (16/08/22): Class introduction & logistics
- week 03 (23/08/22): Reliability techniques & SQLancer
- week 04 (30/08/22): Test oracles I: differential testing
- week 05 (06/09/22): Test oracles II: metamorphic testing
- week 06 (13/09/22): Test oracles III: potpourri
- Recess
- week 07 (27/09/22): Test case generation
- week 08 (04/10/22): TBD
- week 09 (11/10/22): TBD
- week 10 (18/10/22): TBD
- week 11 (25/10/22): TBD
- week 12 (01/11/22): TBD
- week 13 (08/11/22): Backup/project presentations

Volunteers: first two presentations

- Two volunteers needed
- August 30 (two weeks)
- Focus: differential testing
- https://manuelrigger.at/teaching/CS6218/readings.html#differential-testing

Differential Testing

- Data-Oriented Differential Testing of Object-Relational Mapping Systems (ICSE '21) [GitHub] [YouTube]
- Finding bugs in Gremlin-based graph database systems via Randomized differential testing (ISSTA '21) [GitHub] [YouTube]
- DiffStream: Differential Output Testing for Stream Processing Programs (OOPSLA '20) [GitHub] [YouTube]
- APOLLO: Automatic Detection and Diagnosis of Performance Regressions in Database Systems (VLDB '19) [GitHub] [YouTube],

What About You?

- What is your background and/or your (research) interests?
- Why are you taking this class?
- What are your expectations?

https://www.facebook.com/HITZdotmy/posts/10156012835913027

Ensuring the Correctness and Reliability of Data-centric Systems

What are Data-Centric Systems?

- Broad umbrella term
- System in which data is an important asset
- Heterogeneous landscape of systems with partly overlapping functionality

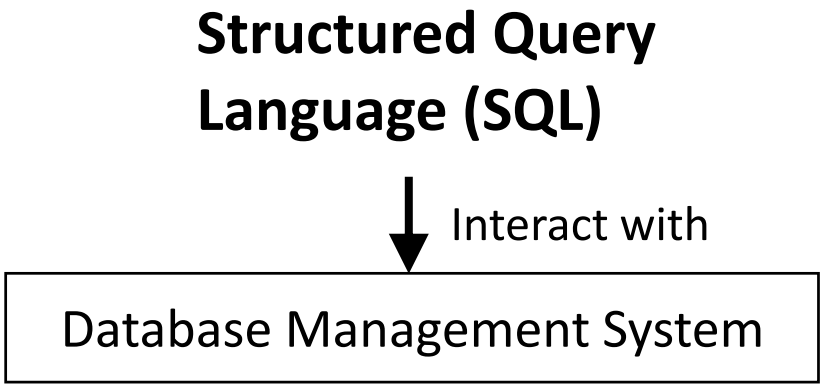
Examples: Database Systems

- Relational database systems
- NoSQL systems
  - Document stores
  - Graph stores
  - Key-value stores
- NewSQL systems

Examples: Database Systems

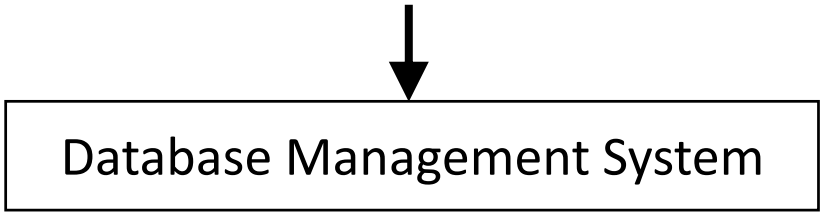
- Relational database systems
- NoSQL systems
  - Document stores
  - Graph stores
  - Key-value stores
- NewSQL systems

Database Management Systems (DBMS)

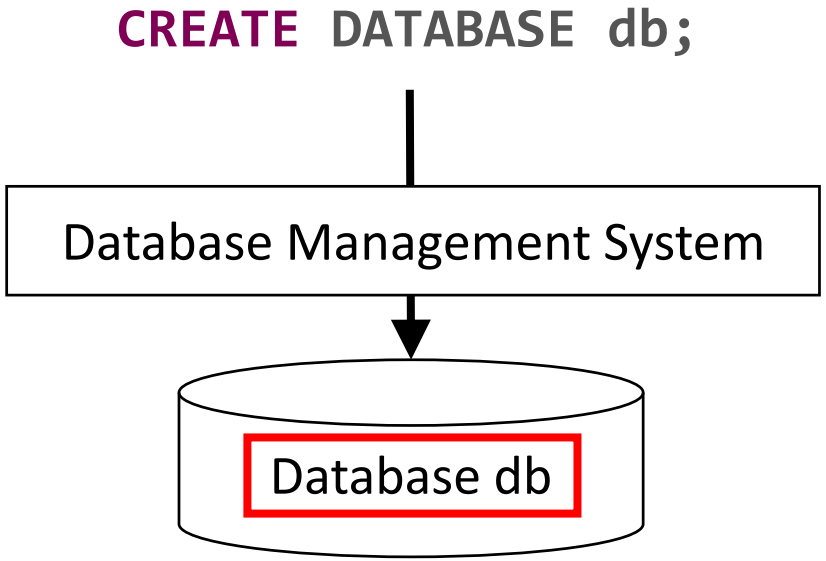


Database Management Systems (DBMS)

```
CREATE DATABASE db;
```

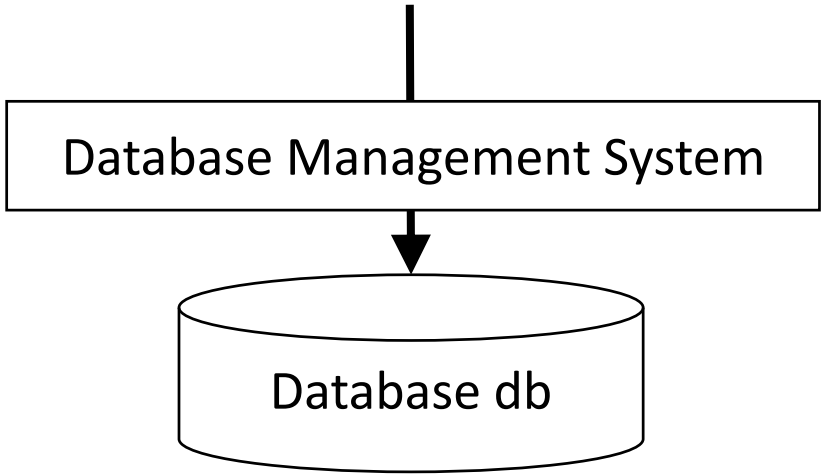


Database Management Systems (DBMS)

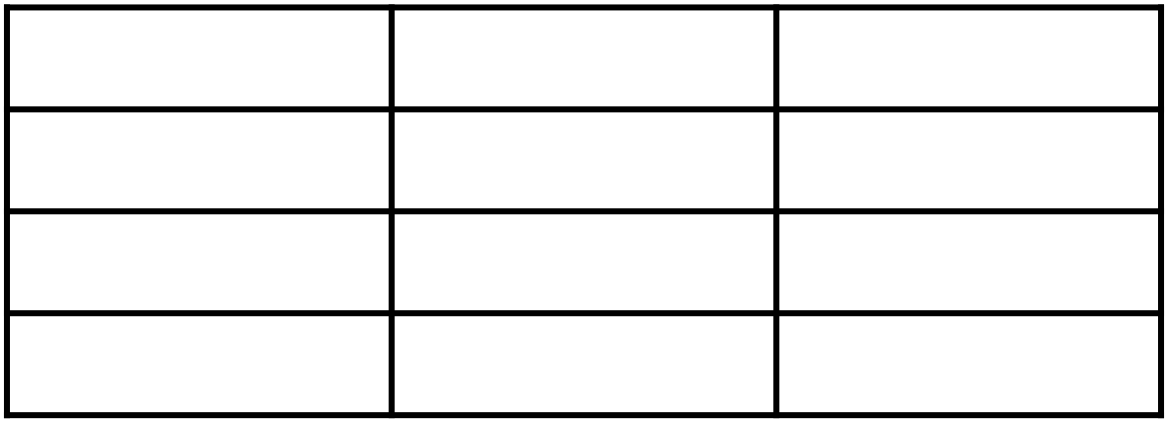


Database Management Systems (DBMS)

```
CREATE TABLE t0(c0 INTEGER);
```

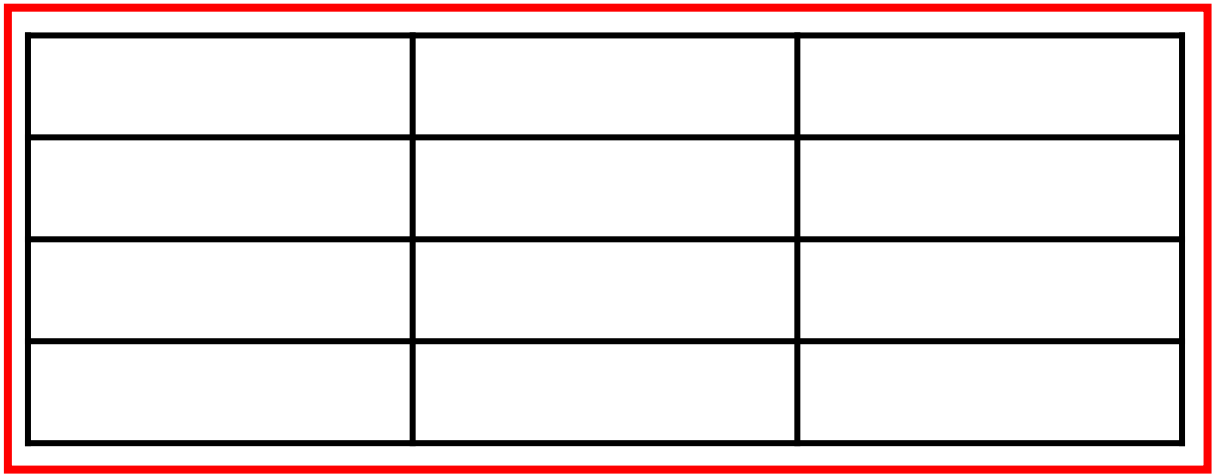


Relational Model



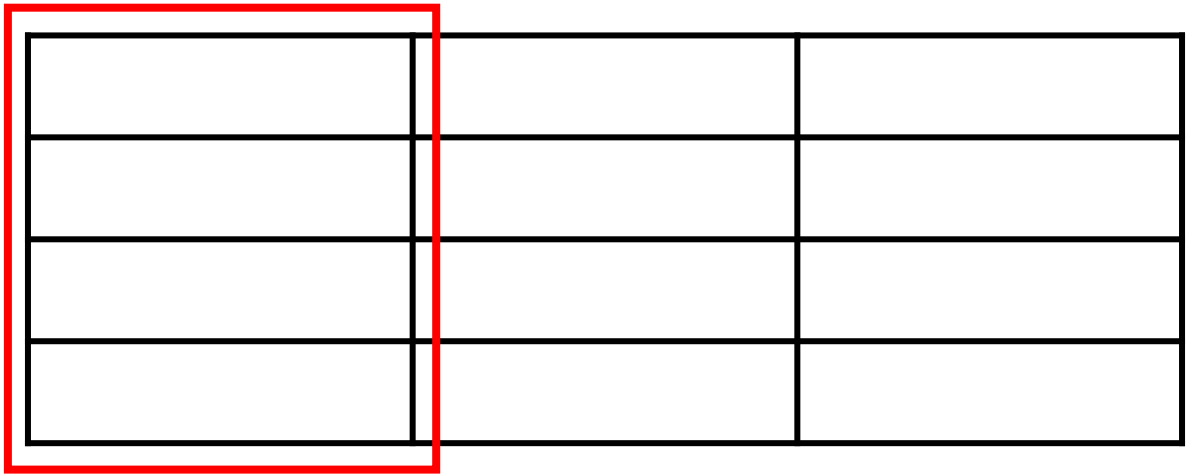
Relational Model

Table or Relation

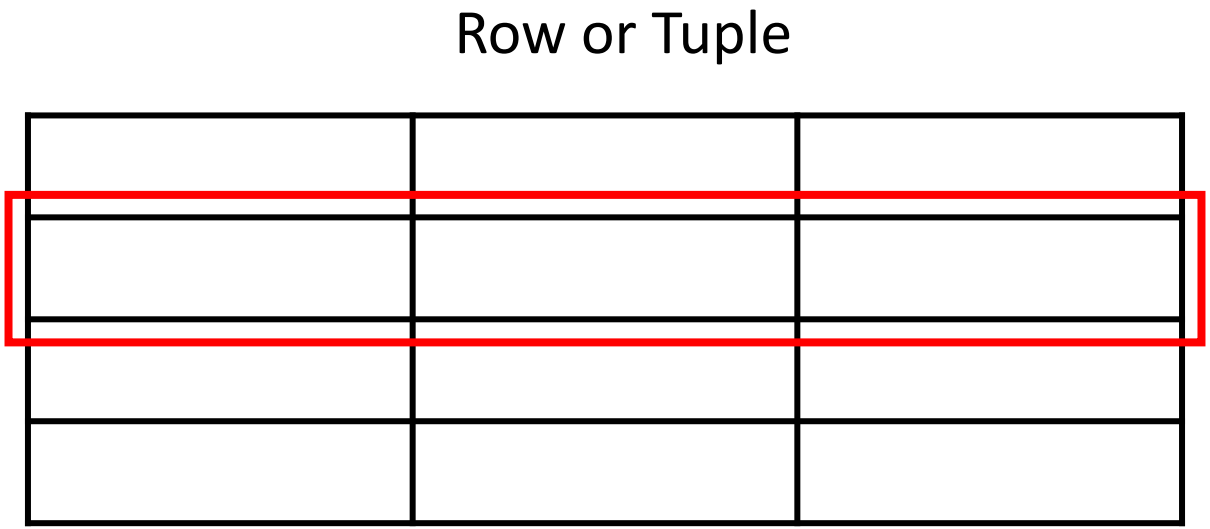


Relational Model

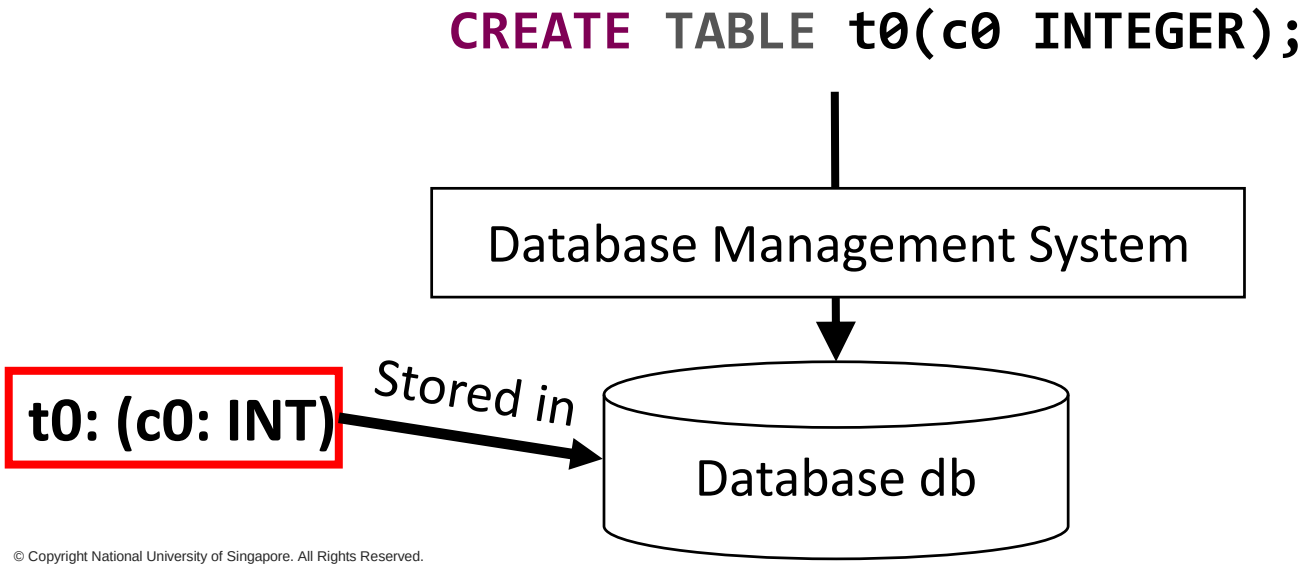
Column or Attribute



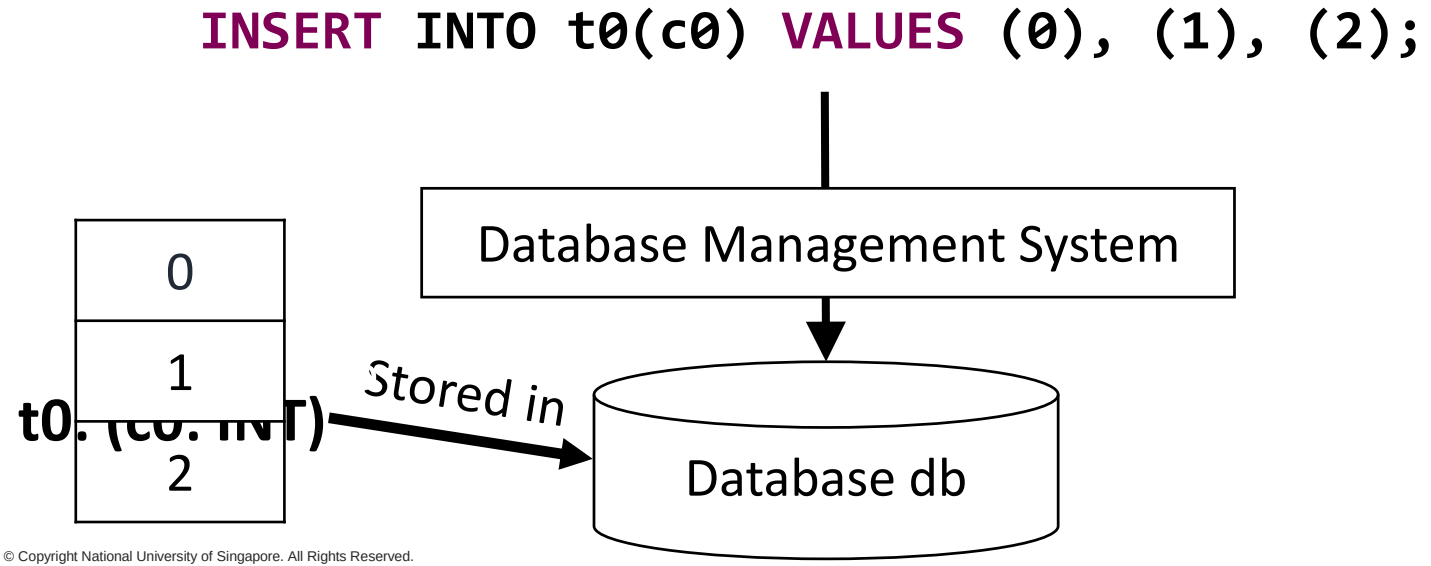
Relational Model



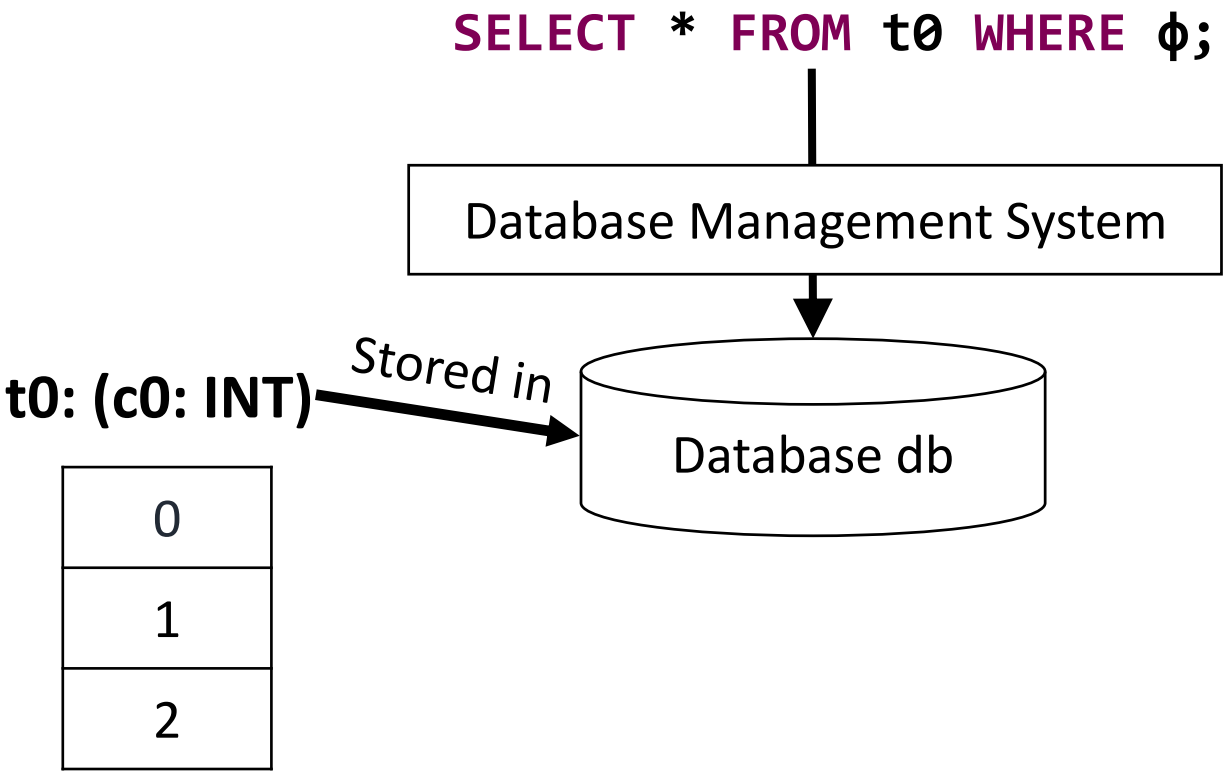
Database Management Systems (DBMS)



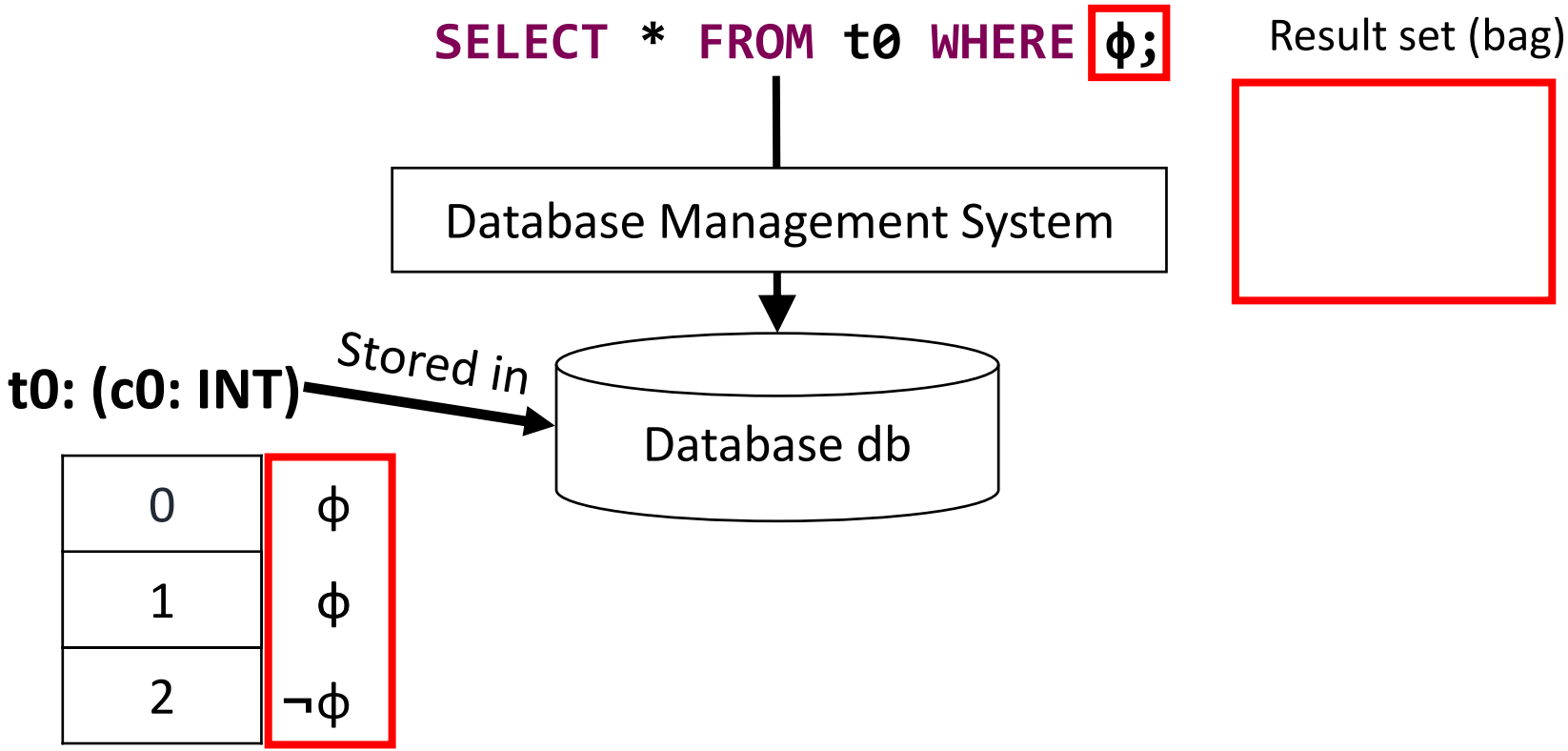
Database Management Systems (DBMS)



Database Management Systems (DBMS)

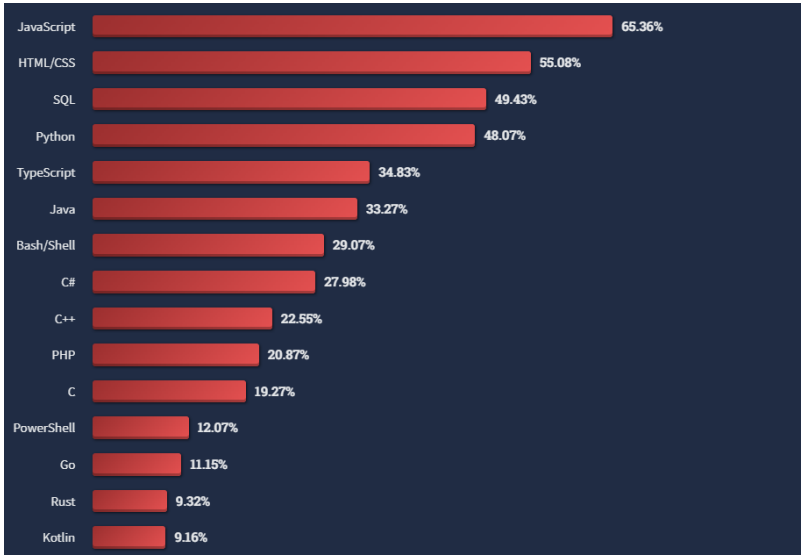


Database Management Systems (DBMS)



Importance of SQL

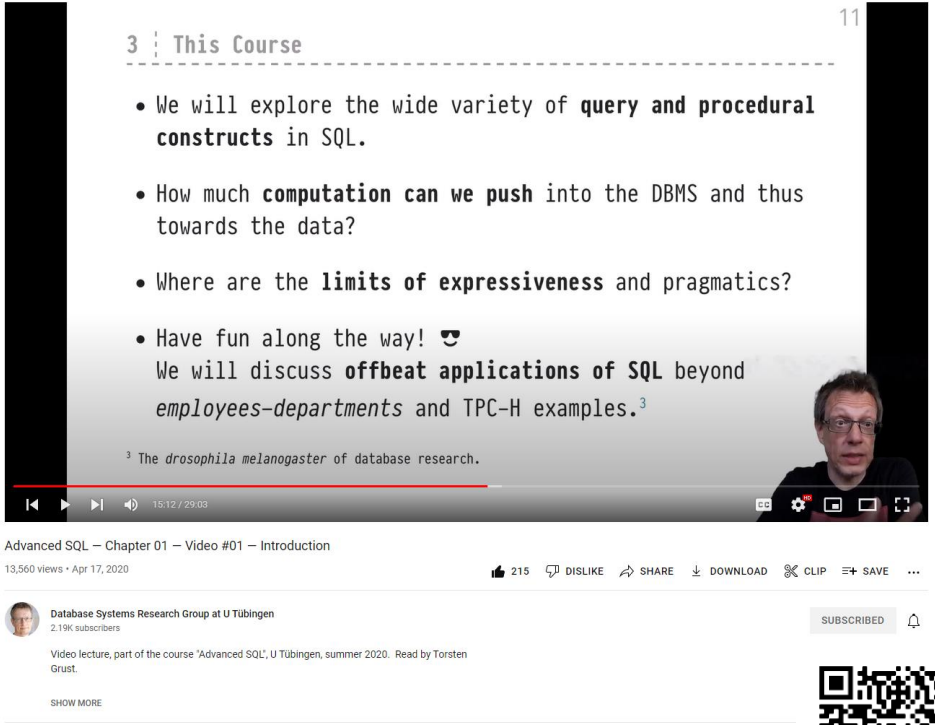
- SQL is one of the most popular programming languages adopted by many data-centric systems
- Pronunciation: Sequel vs S.Q.L



https://survey.stackoverflow.co/2022/#most-popular-technologies-language

Brushing Up Your SQL Knowledge?

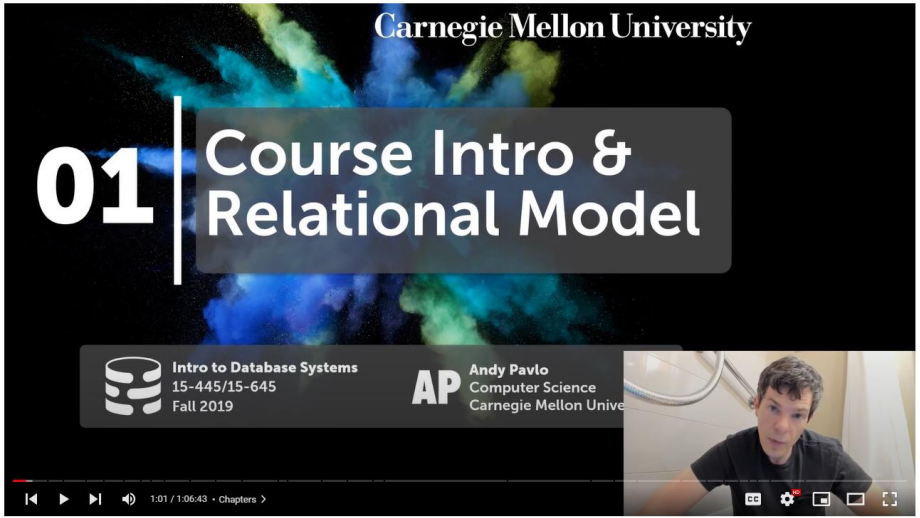
- “Advanced SQL” Lectures
- By Torsten Grust
- Recommended if you want to understand advanced SQL concepts



https://www.youtube.com/playlist?list=PL1XF9qjV8kH12PTd1WfsKeUQU6e83ldfc

Brushing Up Your Database Knowledge?

- “Database Systems” Lectures
- By Andy Pavlo
- Recommend if you want to understand how database systems are implemented



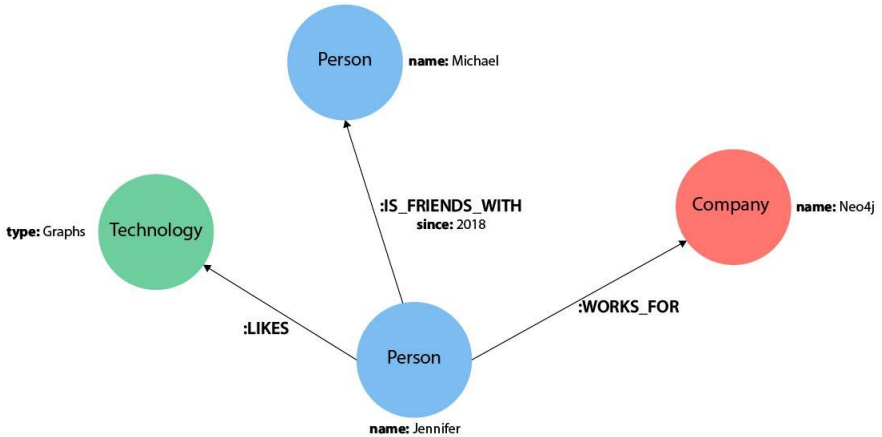
https://www.youtube.com/playlist?list=PLSE8ODhjZXjbohkNBWQs_otTrBTrjyohi

Examples: Database Systems

- Relational database systems
- NoSQL systems
  - Document stores
  - Graph stores
  - Key-value stores
- NewSQL systems

Graph Databases

- Nodes
- Relationships
- Properties



(p:Person {name: "Jennifer"})-[rel:LIKES]->(g:Technology {type: "Graphs"})

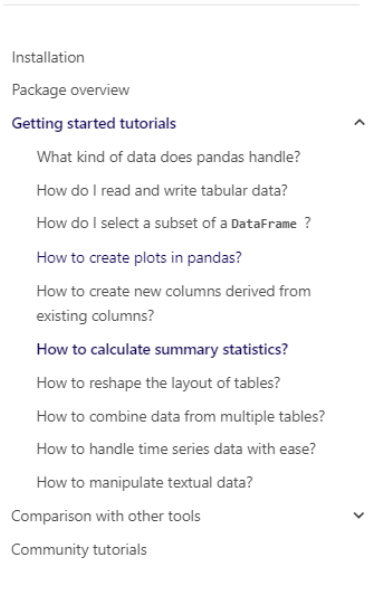
https://neo4j.com/developer/cypher/intro-cypher/

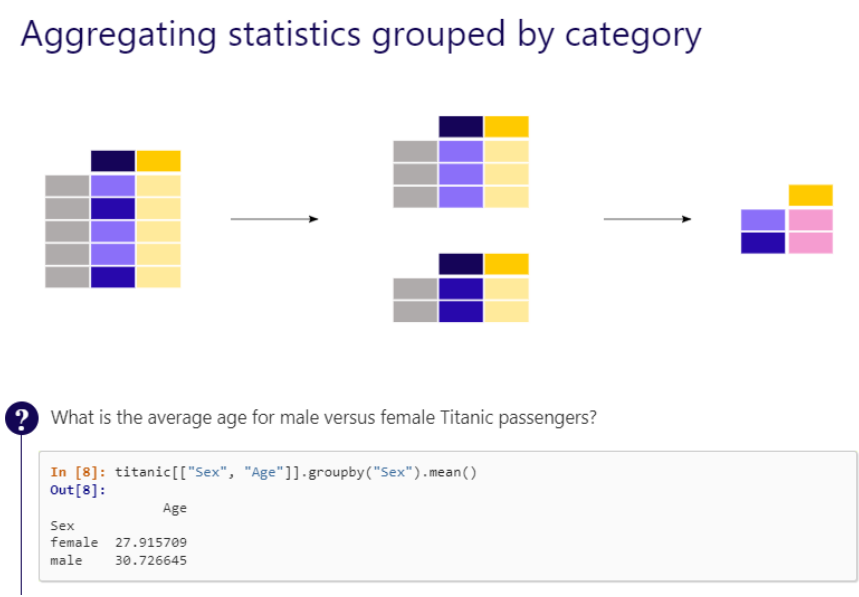
Examples

- Relational database systems
- NoSQL systems
  - Document stores
  - Graph stores
  - Key-value stores
- NewSQL systems
- Data Manipulation Frameworks

Examples: Data Manipulation Frameworks

- Pandas
- dplyr





https://pandas.pydata.org/pandas-docs/stable/getting_started/intro_tutorials/06_calculate_statistics.html

Examples

- Relational database systems
- NoSQL systems
  - Document stores
  - Graph stores
  - Key-value stores
- NewSQL systems
- Data Manipulation Frameworks
- Data Processing Frameworks

Examples: Data Processing

- Apache Spark
- Hadoop
- Apache Flink

In this example, we search through the error messages in a log file.

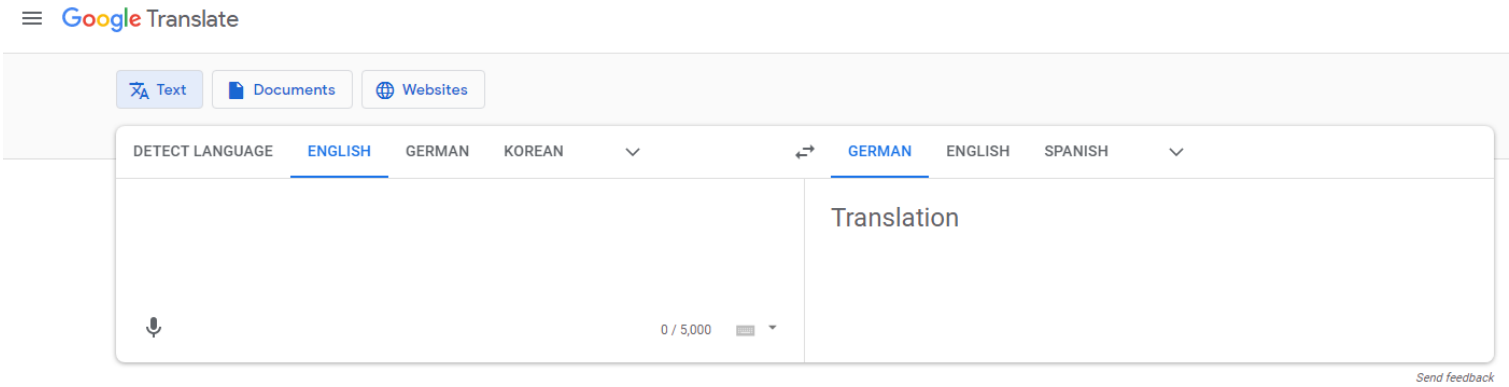
```
val textFile = sc.textFile("hdfs://...") // Creates a DataFrame having a single column named "line“
val df = textFile.toDF("line")
val errors = df.filter(col("line").like("%ERROR%"))
// Counts all the errors
errors.count()
 // Counts errors mentioning MySQL
errors.filter(col("line").like("%MySQL%")).count()
// Fetches the MySQL errors as an array of strings
errors.filter(col("line").like("%MySQL%")).collect()
```


https://spark.apache.org/examples.html

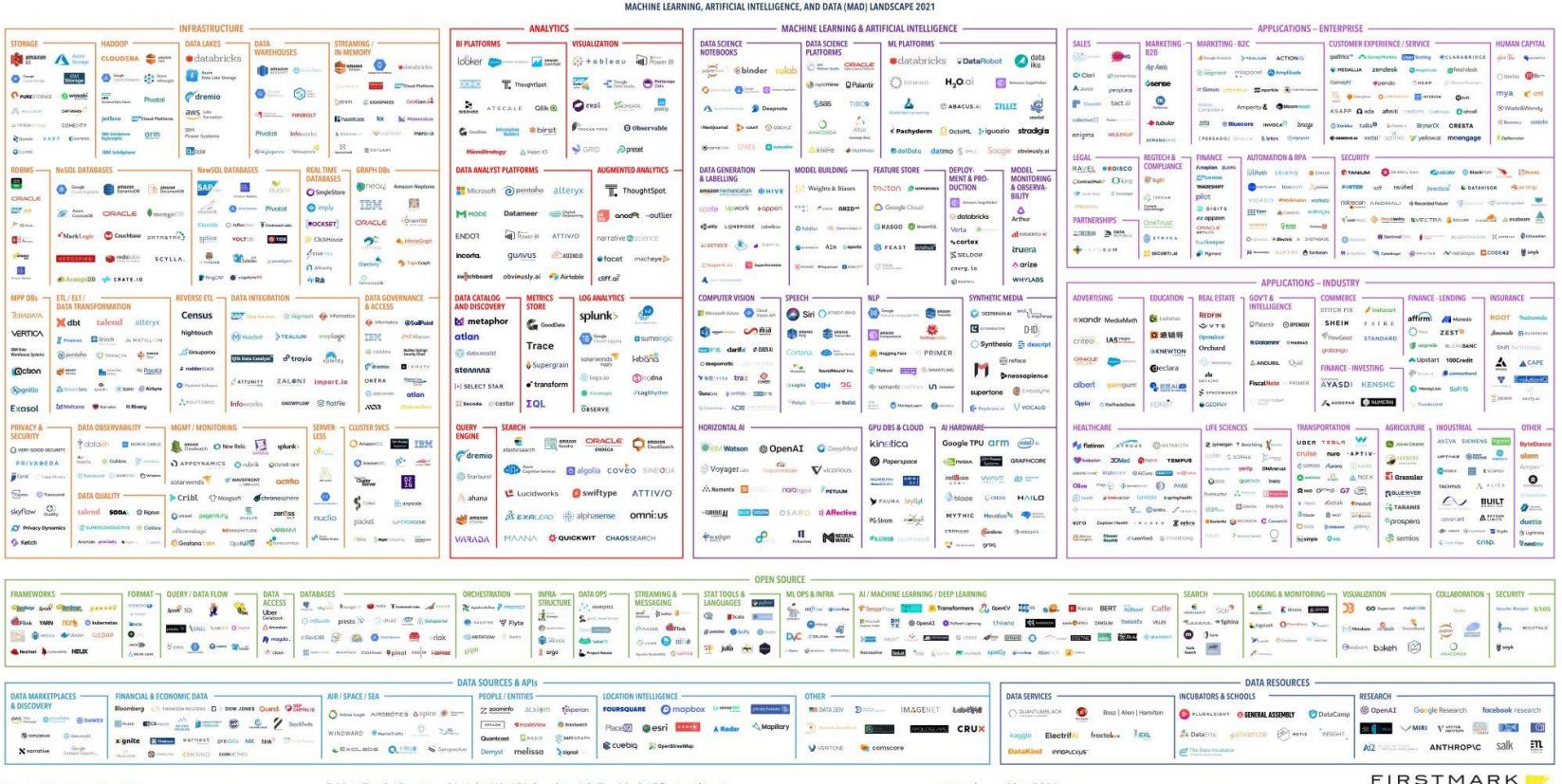
Examples

- Relational database systems
- NoSQL systems
  - Document stores
  - Graph stores
  - Key-value stores
- NewSQL systems
- Data Manipulation Frameworks
- Data Processing Frameworks
- AI Applications

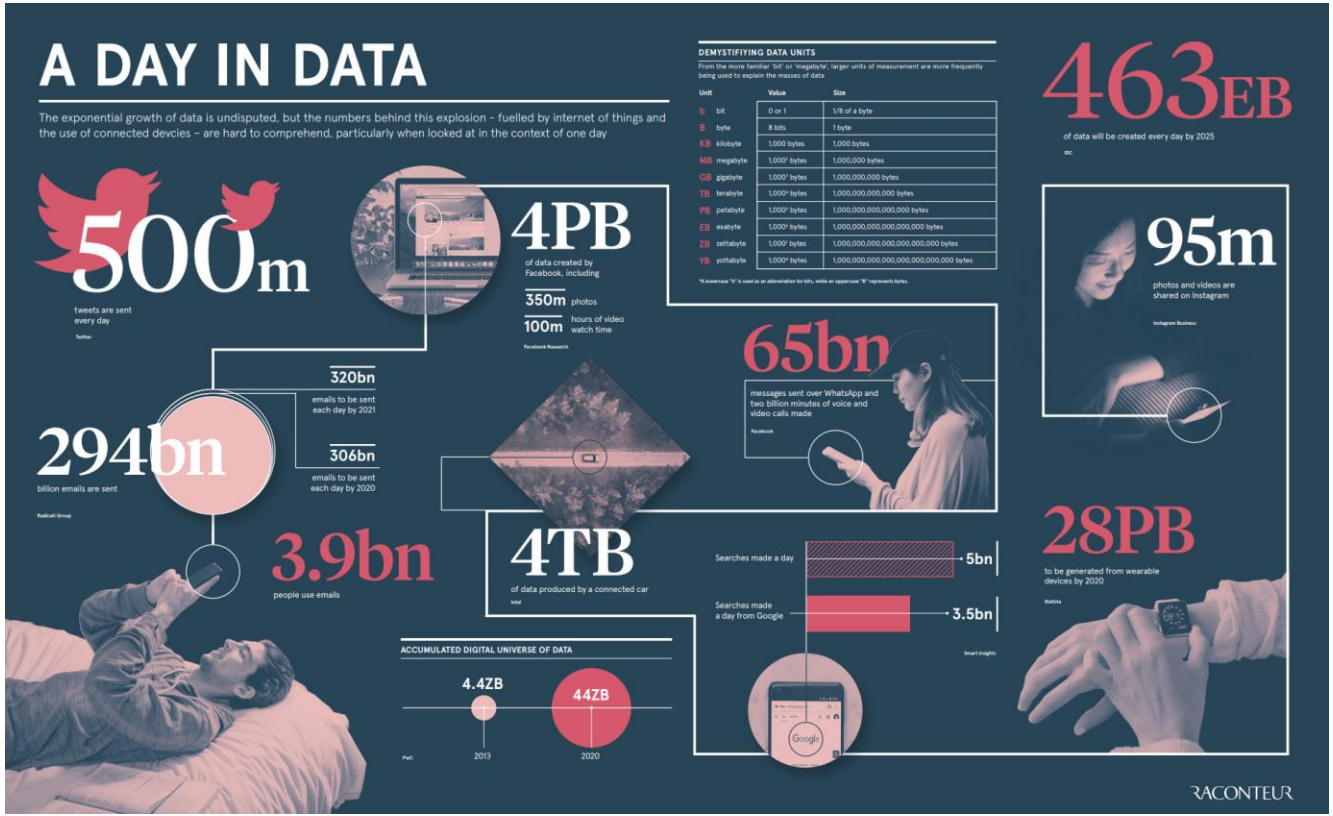
AI Applications: Machine Translation

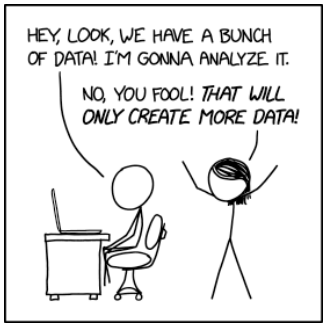


Overview of the Landscape by Matt Turck



Why Are They Important?





https://xkcd.com/2582/

https://www.visualcapitalist.com/wp-content/uploads/2019/04/data-generated-each-day-full.html

Scales of Database Systems

- Embedded databases
- Standard database systems
- Data warehouses and lakehouse systems

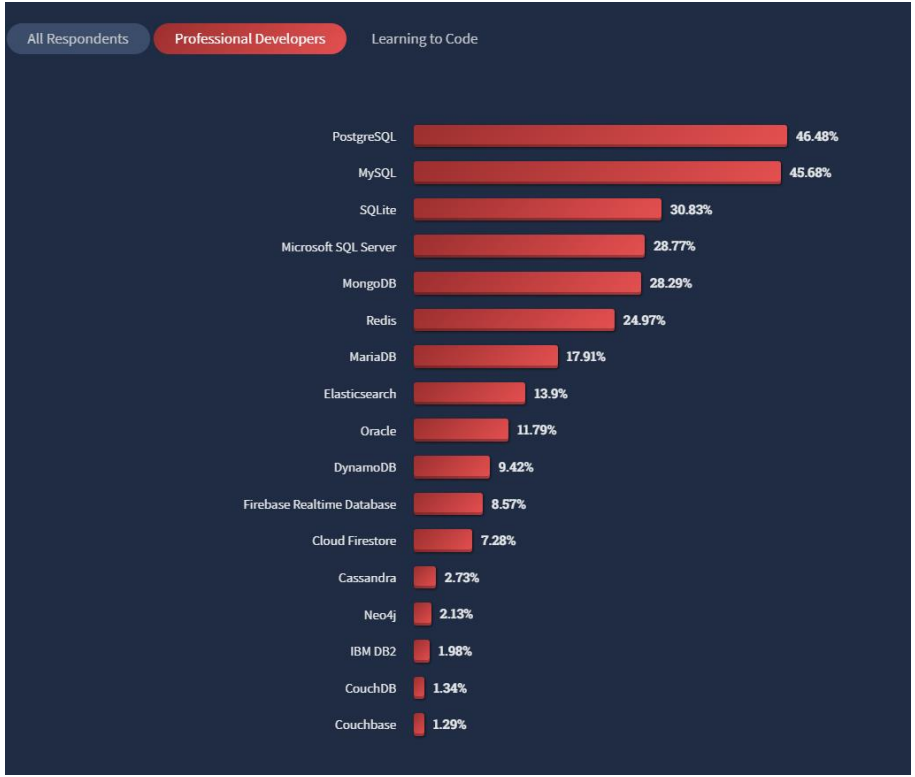
Why Are They Important?

- Embedded databases and edge computing
- Since SQLite is used extensively in every smartphone, and there are more than 4.0 billion (4.0e9) smartphones in active use, each holding hundreds of SQLite database files, it is seems likely that there are over one trillion (1e12) SQLite databases in active use.

https://www.sqlite.org/mostdeployed.html

Why Are They Important?

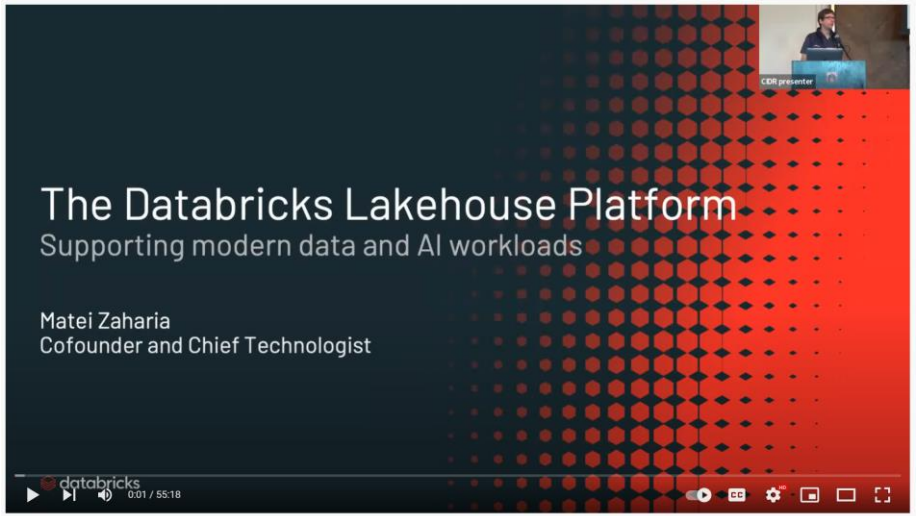
- “Standard” relational database systems like PostgreSQL, MySQL, Microsoft SQL Server, Oracle used by many small and mid-sized companies



https://survey.stackoverflow.co/2022/#most-popular-technologies-database-prof

Why Are They Important?

- Companies store and process large-scale data (petabytes)
- Machine learning and business intelligence solutions
- Cloud solutions



CIDR 2022 Keynote 1: The Databricks Lakehouse Platform by Matei Zaharia

https://www.youtube.com/watch?v=LJb1tR3i9hU

Challenges

- Rapidly-growing data amounts
- Increasing reliance on data-centric solutions
- Heterogeneous landscape of data-centric solutions

Is our data storage and processing infrastructure reliable?

How Reliable is SQLite?

Design process inspired by DO-178B for **safety-critical software systems** in an **aircraft**

SQLite's test cases achieve **100% branch test coverage**

SQLite (~140,000 LOC) has **640 times** as much test code as source code

Anomaly testing (out-of-memory, I/O error, power failures)

https://www.sqlite.org/testing.html

Are These Systems Correct and Reliable?

Motivating Example

t0

c0
0.0

t1

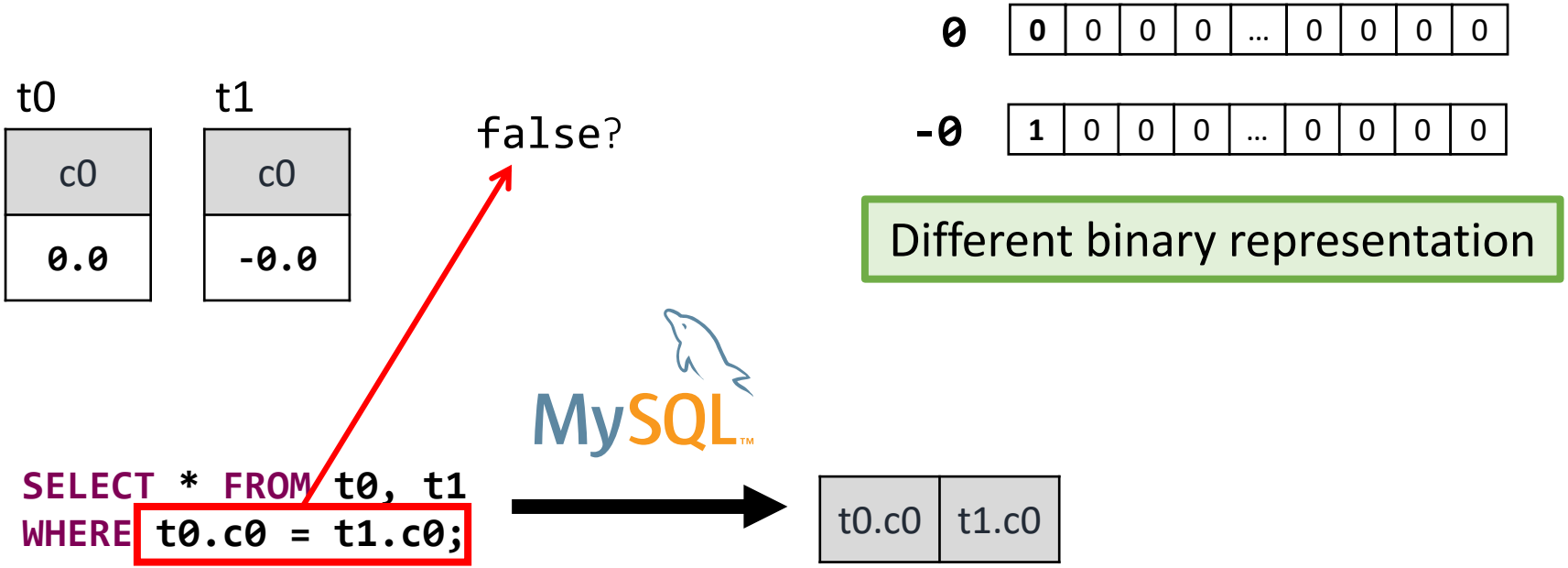
c0
-0.0


```
SELECT * FROM t0, t1
WHERE t0.c0 = t1.c0;
```

→ ?

It might seem **disputable** whether the predicate should evaluate to `true`

Motivating Example



Motivating Example

t0

c0
0.0

t1

c0
-0.0

true?

Evaluates to `true` for **most programming languages**

```
SELECT * FROM t0, t1
WHERE t0.c0 = t1.c0;
```

t0.c0	t1.c0
0	-0

Motivating Example

t0

c0
0.0

t1

c0
-0.0

```
SELECT * FROM t0, t1
WHERE t0.c0 = t1.c0;
```

t0.c0	t1.c0
0	-0

Motivating Example

t0

c0
0.0

t1

c0
-0.0

The latest version of MySQL that we tested failed to fetch the row

```
SELECT * FROM t0, t1
WHERE t0.c0 = t1.c0;
```

t0.c0	t1.c0

https://bugs.mysql.com/bug.php?id=99122

Motivating Example

t0

c0
0.0

t1

c0
-0.0

[3 Apr 2020 13:07] Jon Stephens

Documented fix as follows in the MySQL 8.0.21 changelog:

A query whose predicate compared 0 with -0 where at least one of these was a flaoting-point value returned incorrect results.


```
SELECT * FROM t0, t1
WHERE t0.c0 = t1.c0;
```

t0.c0	t1.c0

https://bugs.mysql.com/bug.php?id=99122

Motivating Example

t0

c0
0.0

t1

c0
-0.0

We could **find the bug without having an accurate understanding** ourselves


```
SELECT * FROM t0, t1
WHERE t0.c0 = t1.c0;
```

t0.c0	t1.c0

https://bugs.mysql.com/bug.php?id=99122

How Can We Ensure Their Reliability?

- Automated testing
  - Differential testing
  - Metamorphic testing
  - Isolation-level testing
- Debugging
- Development environments
- Static & dynamic analysis
- Language design
- Formal verification

Kinds of Bugs

- Crash bugs
- Logic or correctness bugs
- Isolation-level bugs
- Memory management bugs
- Usability issues
- Performance bugs/missed optimization opportunities

Summary

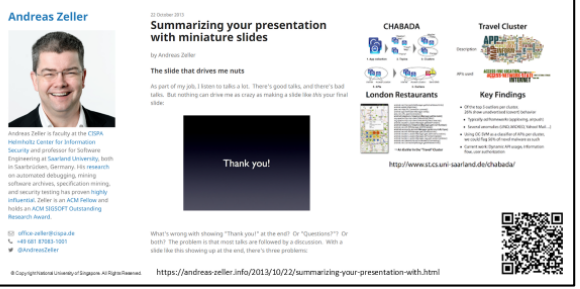
Module Contents

This module will focus on the latest important research in ensuring the correctness, reliability, security, and performance of data-centric systems. It will approach this topic through a software engineering and programming languages lens, providing a broad perspective by considering systems reaching from traditional relational database systems to applications such as machine learning as well as techniques reaching from automated software testing to human-centric approaches.

Grading Scheme

- 40% project
- 40% presentation
- 20% attendance and participation

Designing Presentations



Challenges

- Rapidly-growing data amounts
- Increasing reliance on data-centric solutions
- Heterogeneous landscape of data-centric solutions

Is our data storage and processing infrastructure reliable?

https://manuelrigger.at/teaching/CS6218/