CHENNAI MATHEMATICAL INSTITUTE

Presented at
Qualcomm
on 18th February 2021

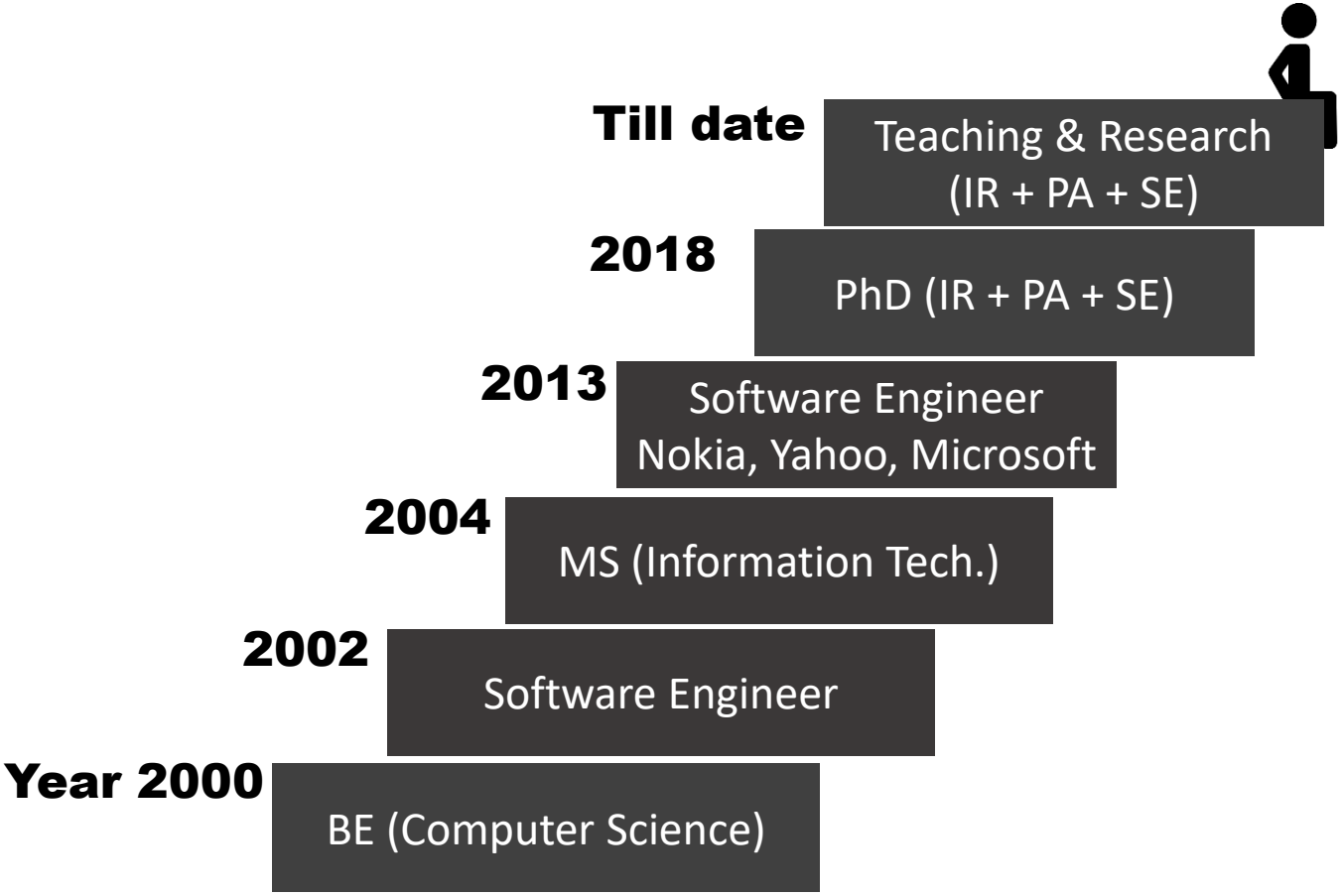
Why should **Software Developers** care for **Mathematics?**

Venkatesh Vinayakarao
venkateshv@cmi.ac.in
http://vvtesh.co.in

Chennai Mathematical Institute

Mathematics, the science of structure, order, and relation that has evolved from elemental practices of counting, measuring, and describing the shapes of objects. **Britannica, https://www.britannica.com/science/mathematics.**

About Me


Till date
Teaching & Research
(IR + PA + SE)
2018
PhD (IR + PA + SE)
2013
Software Engineer
Nokia, Yahoo, Microsoft
2004
MS (Information Tech.)
2002
Software Engineer
Year 2000
BE (Computer Science)

Agenda

Why should **Software Developers** care for **Mathematics**?

Will Discuss	Will not Discuss
✓ Concepts	⦸ Details
✓ Illustrations	⦸ Definitions
✓ Intuitions	⦸ Formalism
✓ Purpose	⦸ Derivations
✓ Properties	⦸ Proofs

Three Stories
Database, Search Engines, Text Processing

Story 1

Evolution of RDBMS and SQL

Image source https://commons.wikimedia.org/wiki/File:Fotothek_df_n-10_0000117.jpg

Interrelated “Data” as a Relation

The Relational Model

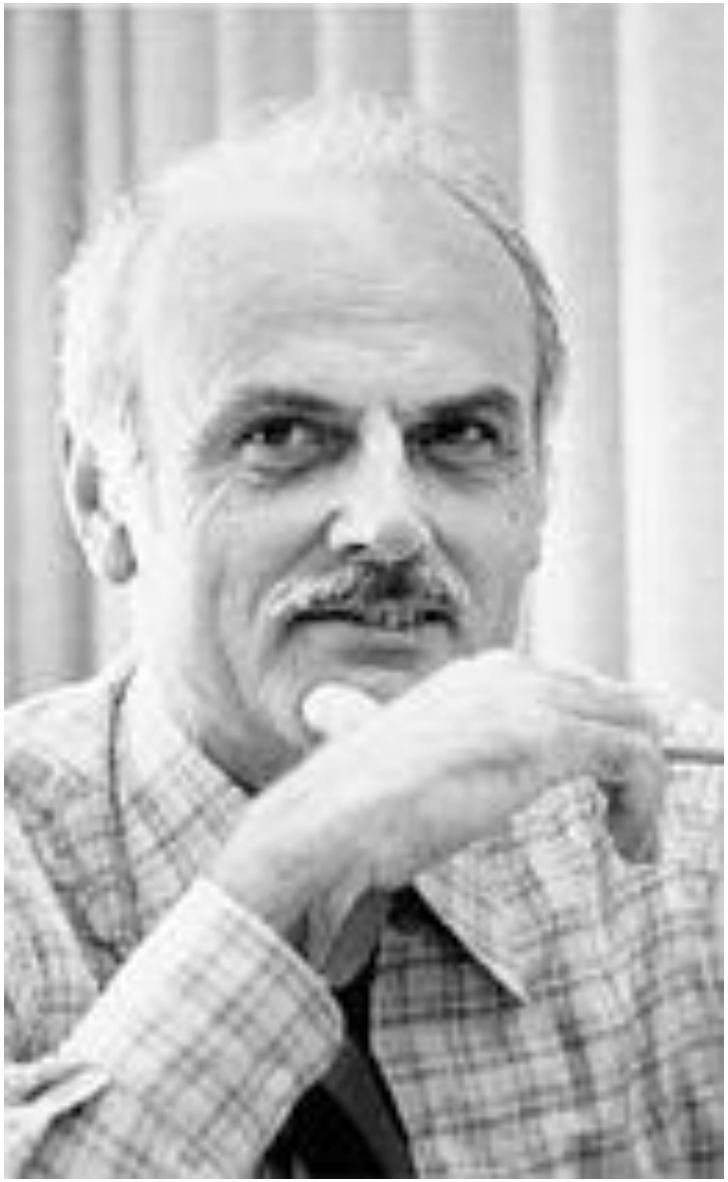
Edgar F. Codd

PhD in Computer Science

Winner of the Turing Award

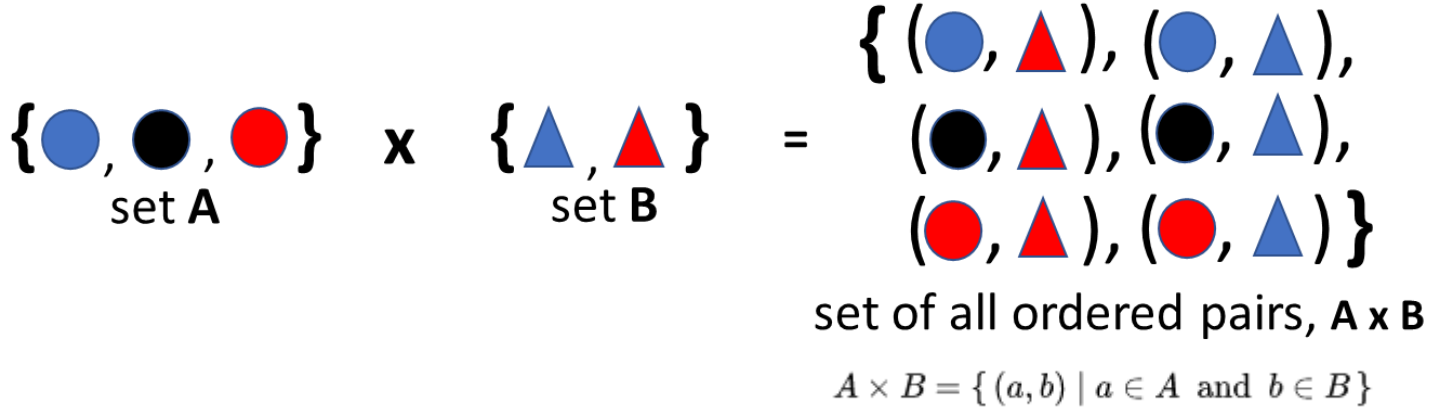
“He made other valuable contributions to computer science, but the relational model, a very influential general theory of data management, remains his most mentioned, analyzed and celebrated achievement.”

– Wikipedia.



The "Relation" from Math Textbook

- A relation R from a set A to set B is a subset of the cartesian product A x B.



Let's say a relation exists between the reds

Relation RedShapes = 

A Relation

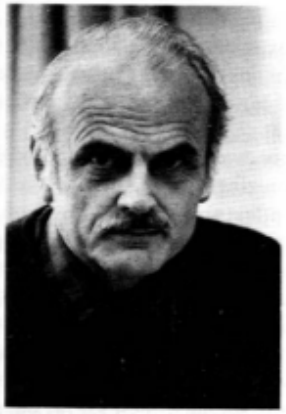
- Let the set, id = {1,2,3}
- Let the set, name = {vv, sd}
- What is id x name?
- We have a relation if we assign a sequential id to each name.

id	name
1	sd
2	vv

id	name
1	sd
1	vv
2	sd
2	vv
3	sd
3	vv

The 1981 ACM Turing Award Lecture

Delivered at ACM '81, Los Angeles, California, November 9, 1981



The 1981 ACM Turing Award was presented to Edgar F. Codd, an IBM Fellow of the San Jose Research Laboratory, by President Peter Denning on November 9, 1981 at the ACM Annual Conference in Los Angeles, California. It is the Association's foremost award for technical contributions to the computing community.

Codd was selected by the ACM General Technical Achievement Award Committee for his "fundamental and continuing contributions to the theory and practice of database management systems." The originator of the relational model for databases, Codd has made further important contributions in the development of relational algebra, relational calculus, and normalization of relations.

Edgar F. Codd joined IBM in 1949 to prepare programs for the Selective Sequence Electronic Calculator. Since then, his work in computing has encompassed logical design of computers (IBM 701 and Stretch), managing a computer center in Canada, heading the development of one of the first operating systems with a general multiprogramming capability, contributing to the logic of self-reproducing automata, developing high level techniques for software specification, creating and extending the relational approach to database management, and developing an English analyzing and synthesizing subsystem for casual users of relational databases. He is also the author of *Cellular Automata*, an early volume in the ACM Monograph Series.

Codd received his B.A. and M.A. in Mathematics from Oxford University in England, and his M.Sc. and Ph.D. in Computer and Communication Sciences from the University of Michigan. He is a Member of the National Academy of Engineering (USA) and a Fellow of the British Computer Society.

The ACM Turing Award is presented each year in commemoration of A. M. Turing, the English mathematician who made major contributions to the computing sciences.

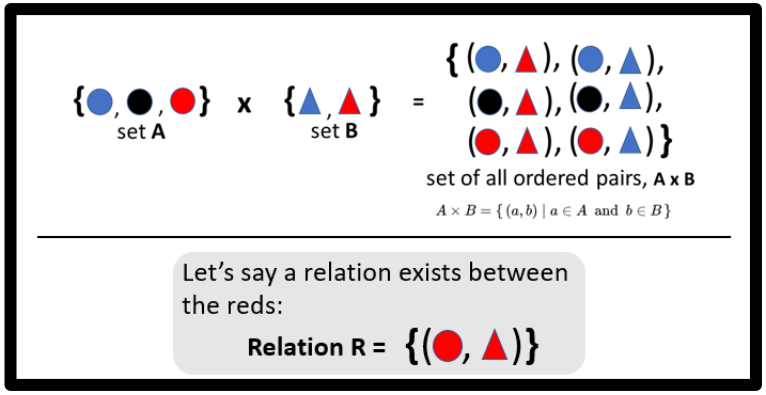
Relational Database: A Practical Foundation for Productivity

E. F. Codd
IBM San Jose Research Laboratory

https://dl.acm.org/doi/pdf/10.1145/1283920.1283937?download=true

Story So Far…

What is a Relation?



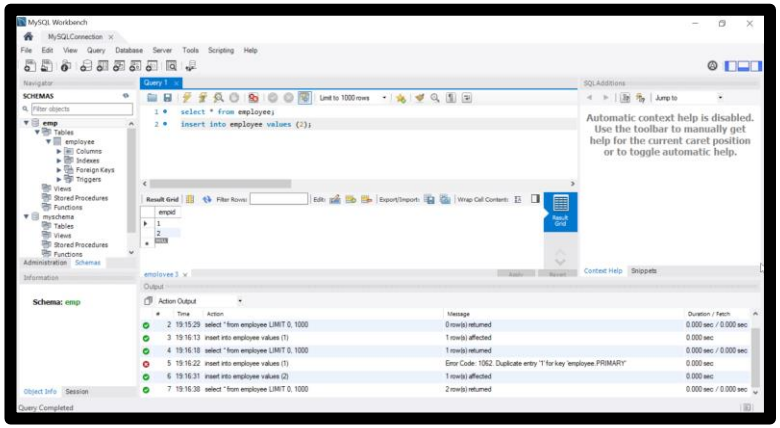
Relational Data Model

id	name
1	sd
1	vv
2	sd
2	vv
3	sd
3	vv

Vs

id	name
1	sd
2	vv

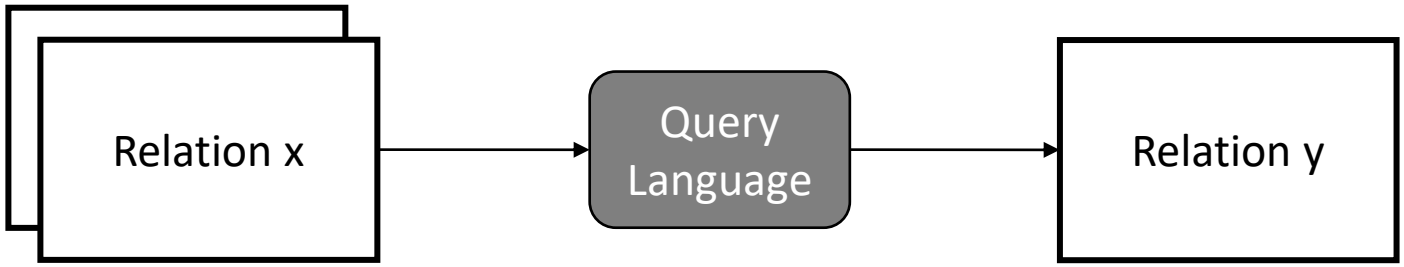
MySQL – An RDBMS



Attendance **relation**
{(1,1), (2,1)}
is same as the table

	studentid	sessionid
▶	1	1
	2	1

Query Languages



Procedural Language
Relational Algebra

Popular Language
SQL

Select Operation

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table

- Relational Algebra

$$\sigma_{\text{dept\_name=“Physics”}}(\text{instructor})$$

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
33456	Gold	Physics	87000

Project Operation

- Notation: $\Pi_{A1, A2,..,Ak}(r)$ where A_i are attribute names

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table

- Example of projection:

$$\Pi_{ID, name, salary}(instructor)$$

Projection

$\Pi_{ID,\ name,\ salary}\ (instructor)$

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table

ID	*name*	*salary*
10101	Srinivasan	65000
12121	Wu	90000
15151	Mozart	40000
22222	Einstein	95000
32343	El Said	60000
33456	Gold	87000
45565	Katz	75000
58583	Califieri	62000
76543	Singh	80000
76766	Crick	72000
83821	Brandt	92000
98345	Kim	80000

Union, Selection and Projection

- $\Pi_{course\_id}(\sigma_{semester=\text{“Fall”} \wedge year=2009}(section)) \cup \Pi_{course\_id}(\sigma_{semester=\text{“Spring”} \wedge year=2010}(section))$

course_id	sec_id	semester	year	building	room_number	time_slot_id
BIO-101	1	Summer	2009	Painter	514	B
BIO-301	1	Summer	2010	Painter	514	A
CS-101	1	Fall	2009	Packard	101	H
CS-101	1	Spring	2010	Packard	101	F
CS-190	1	Spring	2009	Taylor	3128	E
CS-190	2	Spring	2009	Taylor	3128	A
CS-315	1	Spring	2010	Watson	120	D
CS-319	1	Spring	2010	Watson	100	B
CS-319	2	Spring	2010	Taylor	3128	C
CS-347	1	Fall	2009	Taylor	3128	A
EE-181	1	Spring	2009	Taylor	3128	C
FIN-201	1	Spring	2010	Packard	101	B
HIS-351	1	Spring	2010	Painter	514	C
MU-199	1	Spring	2010	Packard	101	D
PHY-101	1	Fall	2009	Watson	100	A

course_id
CS-101
CS-315
CS-319
CS-347
FIN-201
HIS-351
MU-199
PHY-101

Modern day SQL

```
select … from … where …
Union
select … from … where …
```

Story 2

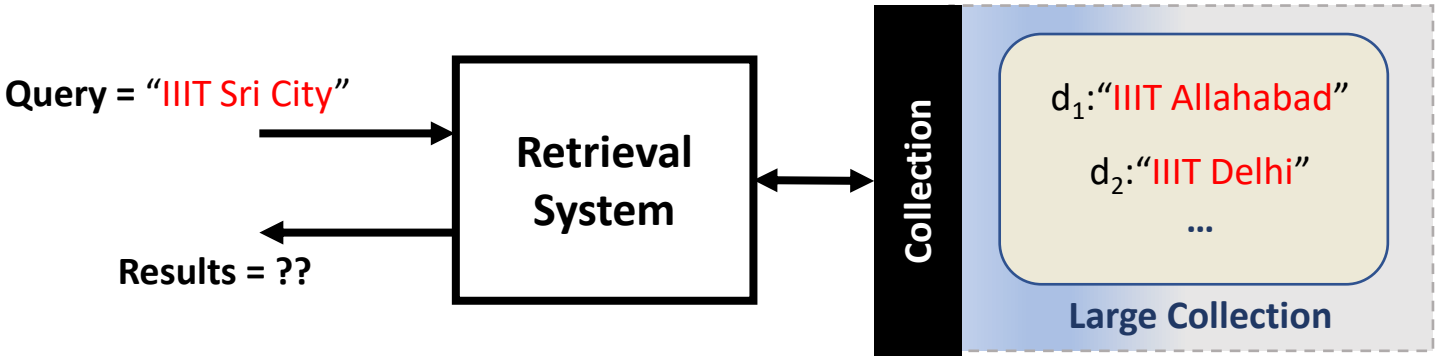
Let us build a search engine!

Image Source
https://philbradley.typepad.com/phil_bradleys_weblog/2016/02/100-search-engines-logos-image-for-you-to-use.html

Simple Retrieval Problem

- Say, we have a **collection** with 5 **documents**, each having the following contents
  - d1: IIIT ALLAHABAD
  - d2: IIIT DELHI
  - d3: IIIT GUWAHATI
  - d4: IIIT KANCHIPURAM
  - d5: IIIT SRI CITY
- Assume, the **Query** is
  - IIIT SRI CITY
- Which **document** will you match and why?

The Problem: How to Build a Retrieval System?



One (bad) Approach

- First match the **term** IIIT.
  - Filter out documents that contain this term.
- Next match the **term** Sri.
  - Filter out documents that contain this term.
- Next match the **term** City.
  - Filter out documents that contain this term.

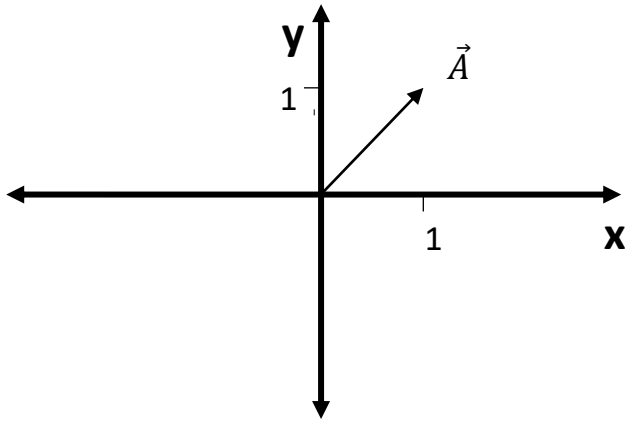
Three iterations!
Quiz: Can we do better?

A Better Approach

Revisiting
Linear Algebra

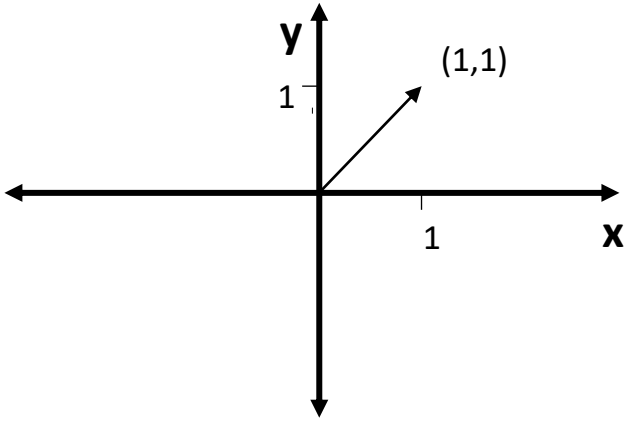
Vector

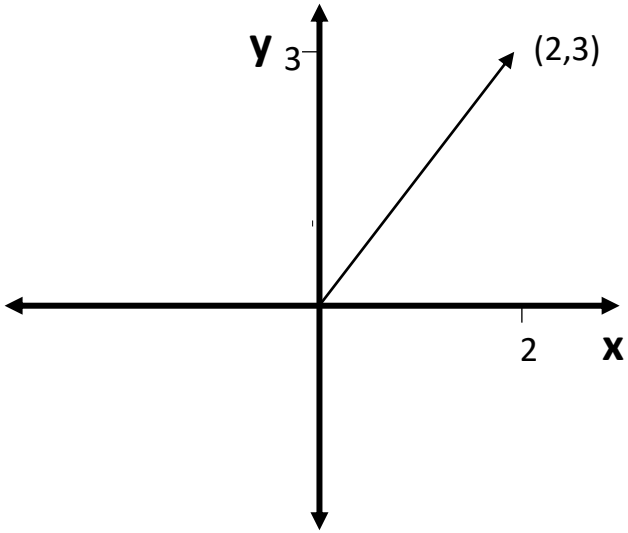
- Geometric entity which has magnitude and direction



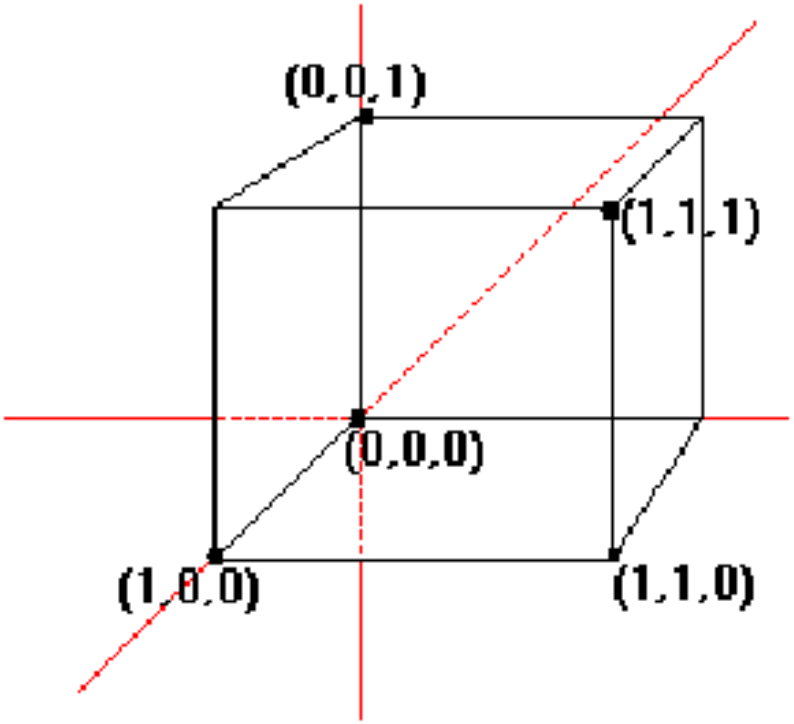
$\vec{A}$ is fixed at (0,0)

How is (2,3) Different?





What is (1,1,1) ?

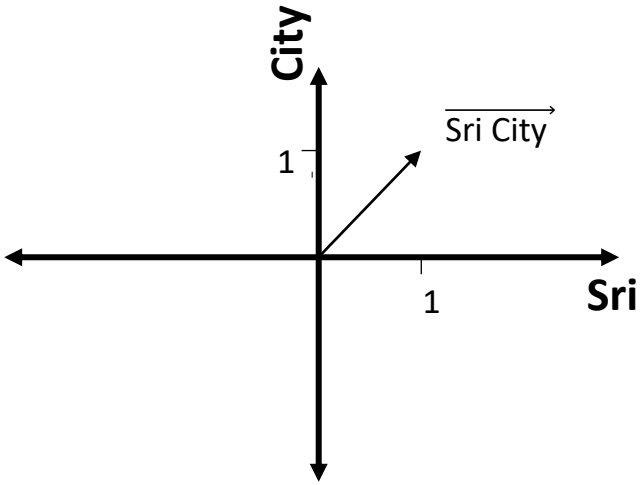


Remember!

A number is just a mathematical object. We give meaning to it!

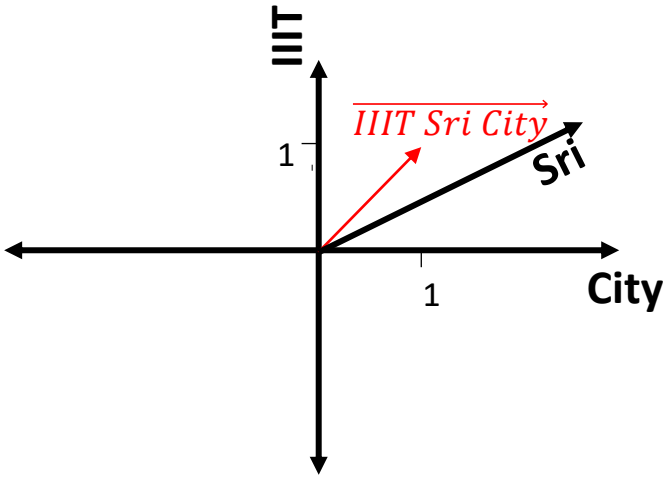
Sentences are Vectors

- “Sri City” as a vector



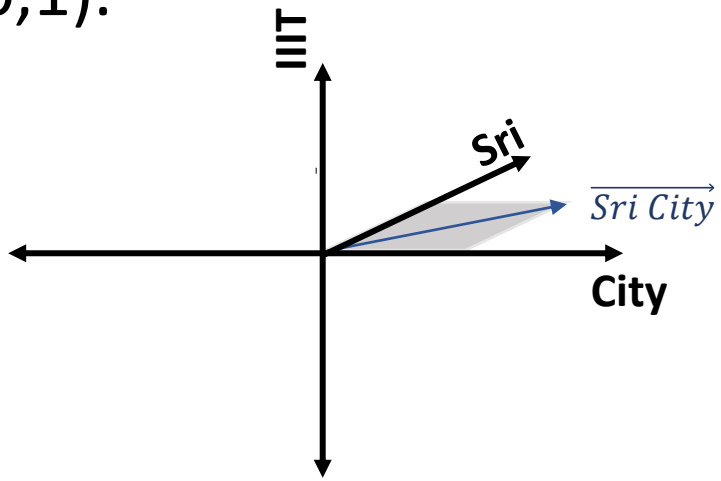
Sentences are Vectors

- "IIIT Sri City" is a 3-dimensional vector



Sentences are Vectors

- On this 3D space, “Sri City” vector will lie on the x (City) and z (Sri) plane. If (x,y,z) denotes the vector, “Sri City” is (1,0,1).



Natural Language Phrases as Vectors

Let query q = "IIIT Sri City".

Let document, d_1 = "IIIT Sri City" and d_2 = "IIIT Delhi".

	IIIT	Sri	City	Delhi
q	1	1	1	0
d_1	1	1	1	0
d_2	1	0	0	1

q = (1,1,1,0), d_1= (1,1,1,0) and d_2 = (1,0,0,1)

Quiz

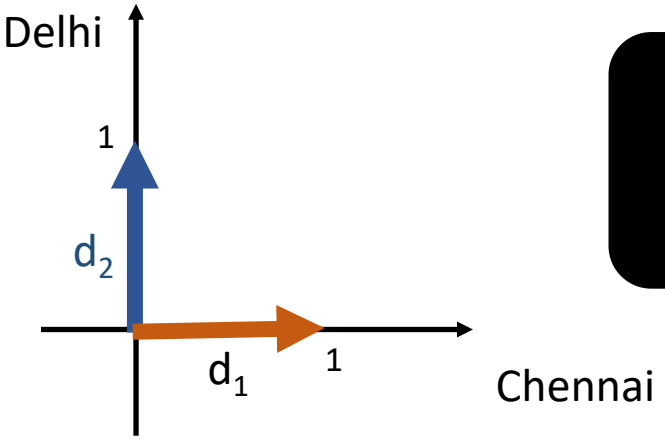
- Considering the following vectors:

	IIIT	Sri	City	Delhi
q	1	1	1	0
d_1	1	1	1	0
d_2	1	0	0	1

- What is the Natural Language (NL) equivalent of (0,1,1,0) ?
- What is the vector for Delhi?

Similarity Score

- Assume, we have the following two documents:
  - d_1 = “Chennai”
  - d_2 = “Delhi”
- On a scale of 0 – 1, how similar are d_1 and d_2?
- What is the angle between d_1 and d_2 vectors?



Can we express similarity as a function of the angles?

0 – 90 to 1 – 0: How?

	0°	30°	45°	60°	90°
sin	0	1/2	1/√2	√3/2	1
cos	1	√3/2	1/√2	1/2	0
tan	0	1/√3	1	√3	Not defined

Back to Trigonometry: Dot Product

- If a and b are non-unit vectors, what is the cosine of angle between them (cos Θ)?

$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \, \|\mathbf{b}\| \cos(\theta)$$

(or)

$$\cos(\theta) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \, \|\mathbf{b}\|}$$

Example

Let query q = "BITS Pilani".

Let document, d_1 = "BITS Pilani Goa Campus" and d_2 = "IIIT Delhi".

	BITS	Pilani	Goa	Campus	IIIT	Delhi
q	1	1	0	0	0	0
d_1	1	1	1	1	0	0
d_2	0	0	0	0	1	1

In our VSM, q = (1,1,0,0,0,0), d_1= (1,1,1,1,0,0) and d_2 = (0,0,0,0,1,1)

similarity(d_1, q) = $\frac{d_1.q}{||d_1||\,||q||} = \frac{1.1 + 1.1}{\sqrt{1^2+1^2+1^2+1^2}\,\sqrt{1^2+1^2}} = 0.71.$

similarity(d_2, q) = $\frac{d_2.q}{||d_2||\,||q||} = 0.$

An Assumption

More frequent appearance of a query term implies higher document relevance.

Which of the Following are Sets?

- {1, 2, 3, 4, 5, 6, 5, 7, 8, 9, 10, 11, 12, 13}
- {A, B, C, D, E, F, G, H, I, I, J, K, L, M, N, O}
- {apple, banana, orange, apple, banana, orange}

Vadivelu, a popular tamil actor

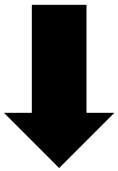
Bag

- {1, 2, 3, 4, 5, 6, 5, 7, 8, 9, 10, 11, 12, 13}
- {A, B, C, D, E, F, G, H, I, I, J, K, L, M, N, O}
- {apple, banana, orange, apple, banana, orange}

Set of Words Representation

- "IIIT Sri City" → {IIIT, Sri, City}
- "IIIT Sri City, Sri City" → {IIIT, Sri, City}

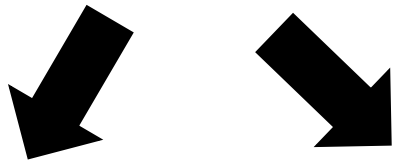
(Assuming, we ignore the punctuations)



	IIIT	Sri	City
q	1	1	1

Bag of Words Representation

- "IIIT Sri City" → {IIIT, Sri, City}
- "IIIT Sri City, Sri City" → [IIIT, Sri, Sri, City, City]



IIIT Sri City

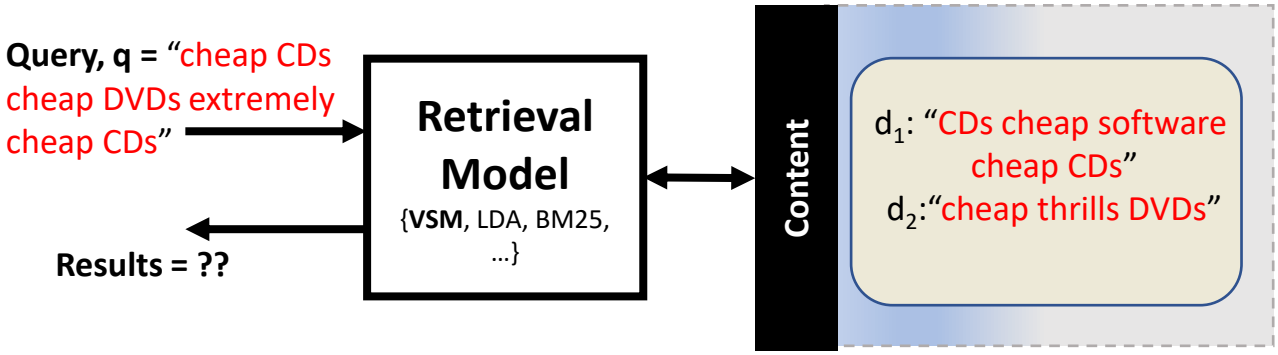
	IIIT	Sri	City
q	1	1	1

IIIT Sri City, Sri City

	IIIT	Sri	City
q	1	2	2

Leads to different vectors

Which Document to Retrieve?



	cheap	CDs	DVDs	extremely	software	thrills
q	3	2	1	1	0	0
d_1	2	2	0	0	1	0
d_2	1	0	1	0	0	1

$sim(q,d_1) = 0.86$

$sim(q,d_2) = 0.59$

Example taken from Manning's IR Book (https://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf)

Story 3

We know a lot about Trees!

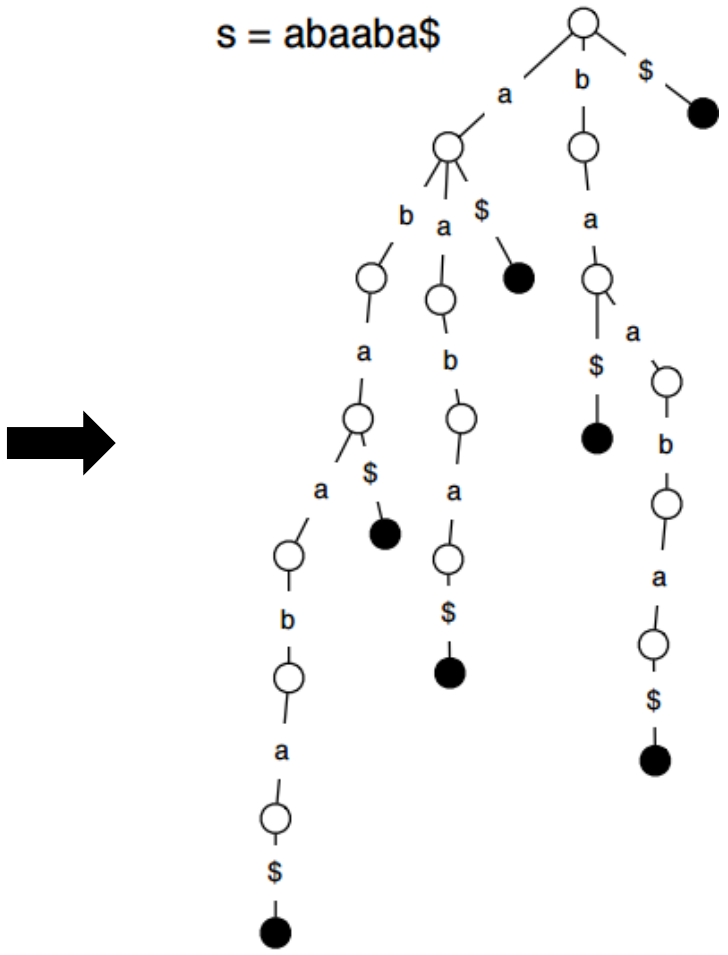
Photo by Scott Graham on Unsplash

Popular Interview Questions

- How can you find if a given string S is a substring of another string T?
- How can you find the number of times S occurs in T?
- Is S a suffix of T?
- Find the longest repeating substring of T.
- Given two strings X and Y, find the longest common substring of X and Y.

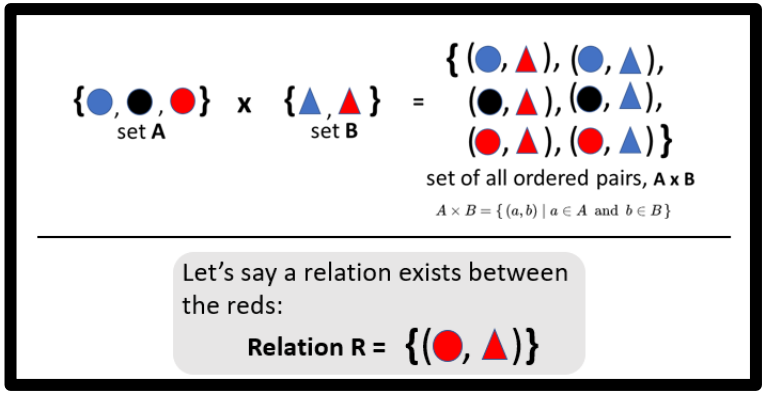
Processing Strings with Suffix Trees

- How can you find if a given string S is a substring of another string T?
- How can you find the number of times S occurs in T?
- Is S a suffix of T?
- Find the longest repeating substring of T.
- Given two strings X and Y, find the longest common substring of X and Y.



Summary: Relational Model

What is a Relation?



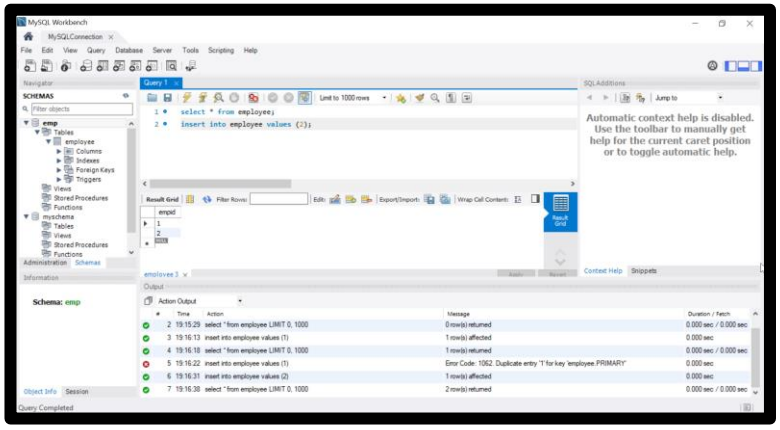
Relational Data Model

id	name
1	sd
1	vv
2	sd
2	vv
3	sd
3	vv

Vs

id	name
1	sd
2	vv

MySQL – An RDBMS

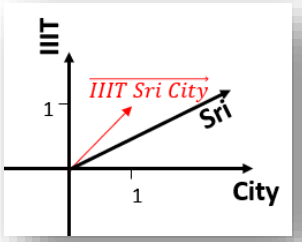


Attendance **relation**
{(1,1), (2,1)}
is same as the table

	studentid	sessionid
▶	1	1
	2	1

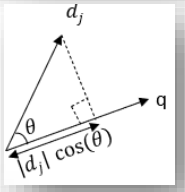
Summary: Vector Space Model

Model Documents as Vectors



Compute Similarity

$$\cos(\theta) = \frac{d_j . q}{||d_j|| \, ||q||}$$



A Worked-Out Example

Query, q = "cheap CDs cheap DVDs extremely cheap CDs"

Retrieval Model {VSM, LDA, BM25, ...}

Results = ??

Content

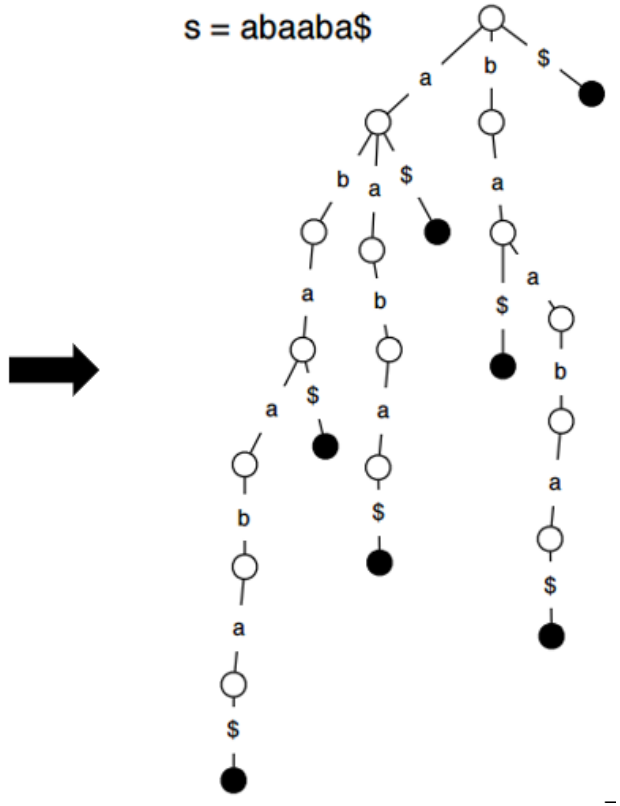
d_1: "CDs cheap software cheap CDs"

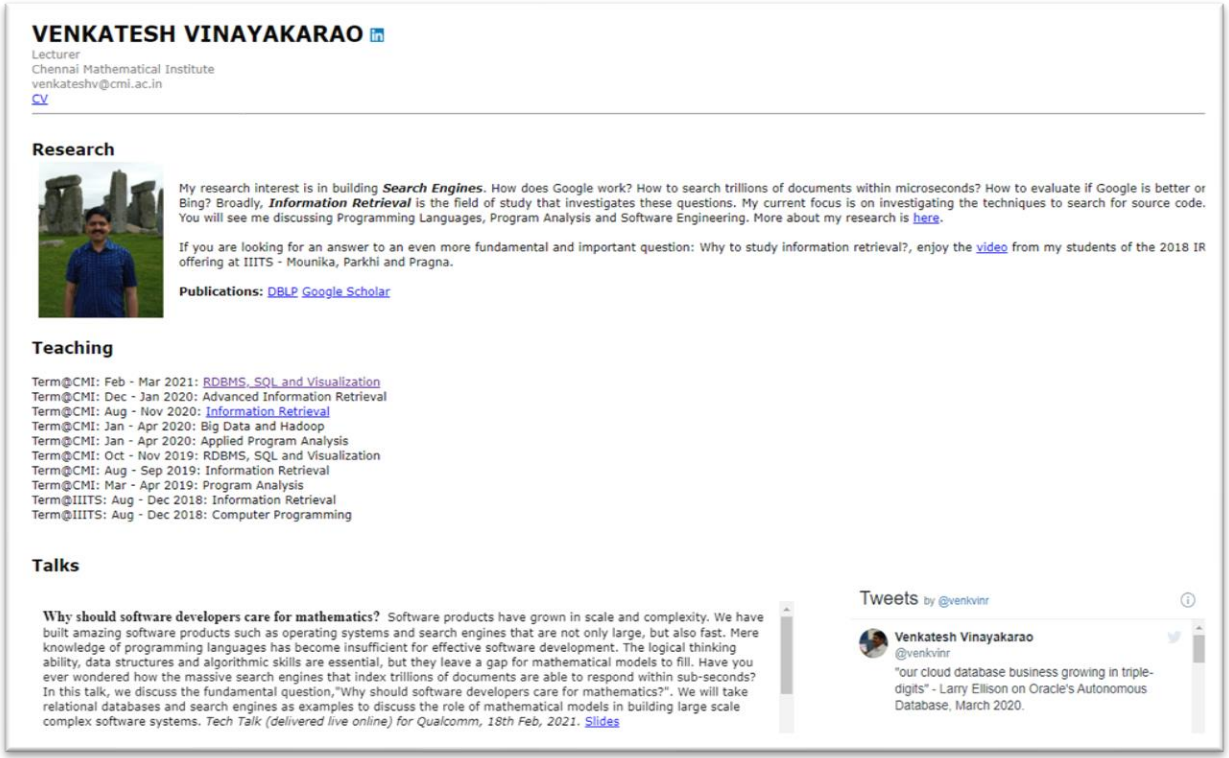
d_2:"cheap thrills DVDs"

	cheap	CDs	DVDs	extremely	software	thrills	
q	3	2	1	1	0	0	
d_1	2	2	0	0	1	0	← $sim(q,d_1) = 0.86$
d_2	1	0	1	0	0	1	← $sim(q,d_2) = 0.59$

Summary: Suffix Trees

- How can you find if a given string S is a substring of another string T?
- How can you find the number of times S occurs in T?
- Is S a suffix of T?
- Find the longest repeating substring of T.
- Given two strings X and Y, find the longest common substring of X and Y.





VENKATESH VINAYAKARAO

Lecturer
Chennai Mathematical Institute
venkateshv@cmi.ac.in
CV

Research

My research interest is in building ***Search Engines***. How does Google work? How to search trillions of documents within microseconds? How to evaluate if Google is better or Bing? Broadly, ***Information Retrieval*** is the field of study that investigates these questions. My current focus is on investigating the techniques to search for source code. You will see me discussing Programming Languages, Program Analysis and Software Engineering. More about my research is here.

If you are looking for an answer to an even more fundamental and important question: Why to study information retrieval?, enjoy the video from my students of the 2018 IR offering at IIITS - Mounika, Parkhi and Pragna.

Publications: DBLP Google Scholar

Teaching

Term@CMI: Feb - Mar 2021: RDBMS, SQL and Visualization
Term@CMI: Dec - Jan 2020: Advanced Information Retrieval
Term@CMI: Aug - Nov 2020: Information Retrieval
Term@CMI: Jan - Apr 2020: Big Data and Hadoop
Term@CMI: Jan - Apr 2020: Applied Program Analysis
Term@CMI: Oct - Nov 2019: RDBMS, SQL and Visualization
Term@CMI: Aug - Sep 2019: Information Retrieval
Term@CMI: Mar - Apr 2019: Program Analysis
Term@IIITS: Aug - Dec 2018: Information Retrieval
Term@IIITS: Aug - Dec 2018: Computer Programming

Talks

Why should software developers care for mathematics? Software products have grown in scale and complexity. We have built amazing software products such as operating systems and search engines that are not only large, but also fast. Mere knowledge of programming languages has become insufficient for effective software development. The logical thinking ability, data structures and algorithmic skills are essential, but they leave a gap for mathematical models to fill. Have you ever wondered how the massive search engines that index trillions of documents are able to respond within sub-seconds? In this talk, we discuss the fundamental question,"Why should software developers care for mathematics?". We will take relational databases and search engines as examples to discuss the role of mathematical models in building large scale complex software systems. *Tech Talk (delivered live online) for Qualcomm, 18th Feb, 2021.* Slides

Tweets by @venkvinr

Venkatesh Vinayakarao
@venkvinr
"our cloud database business growing in triple-digits" - Larry Ellison on Oracle's Autonomous Database, March 2020.

Thank You

venkateshv@cmi.ac.in

This slide deck is available at http://vvtesh.co.in/.