Module 0

History of Databases

Database System

What would be **your** definition?

Database System

A commonly accepted definition would be

> **Database**: a system that can handle *arbitrary* datasets

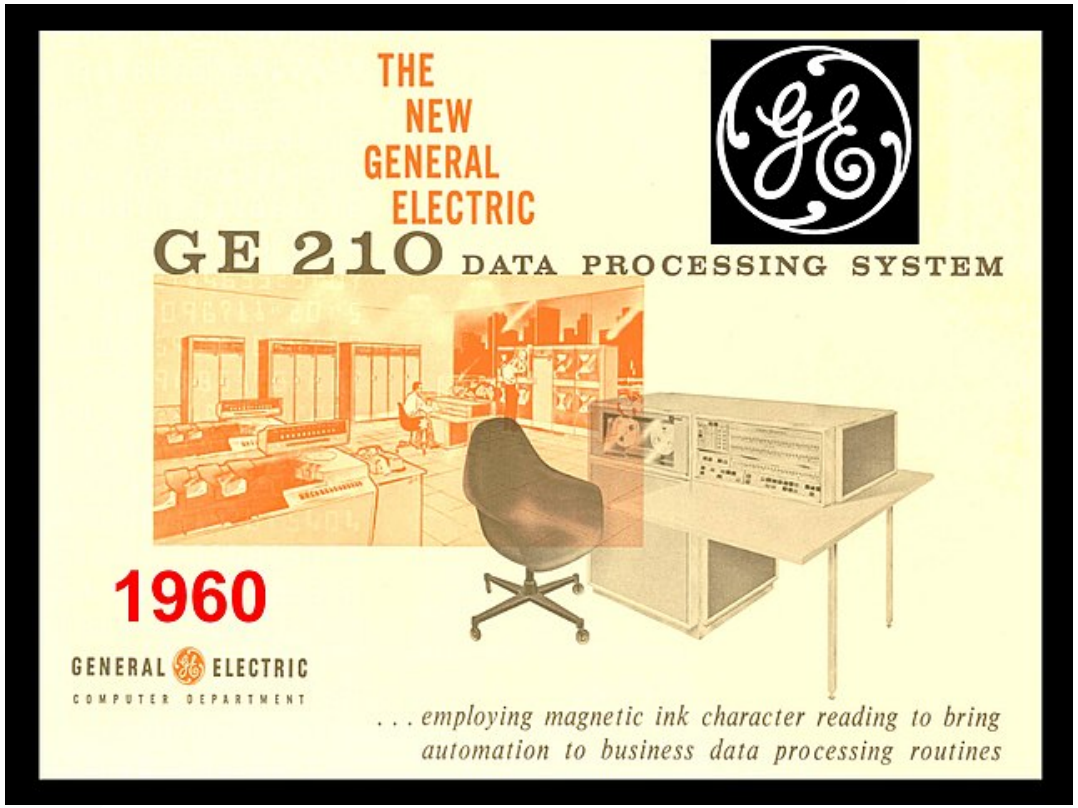
Any idea in which **decade** those were invented?

1960s - IDS

Charles Bachman joins **General Electric** to work on what is commonly accepted as the first ever **DBMS** (**D**ata**b**ase **M**anagement **S**ystem) named **Integrated Data Store** a.k.a. **IDS**

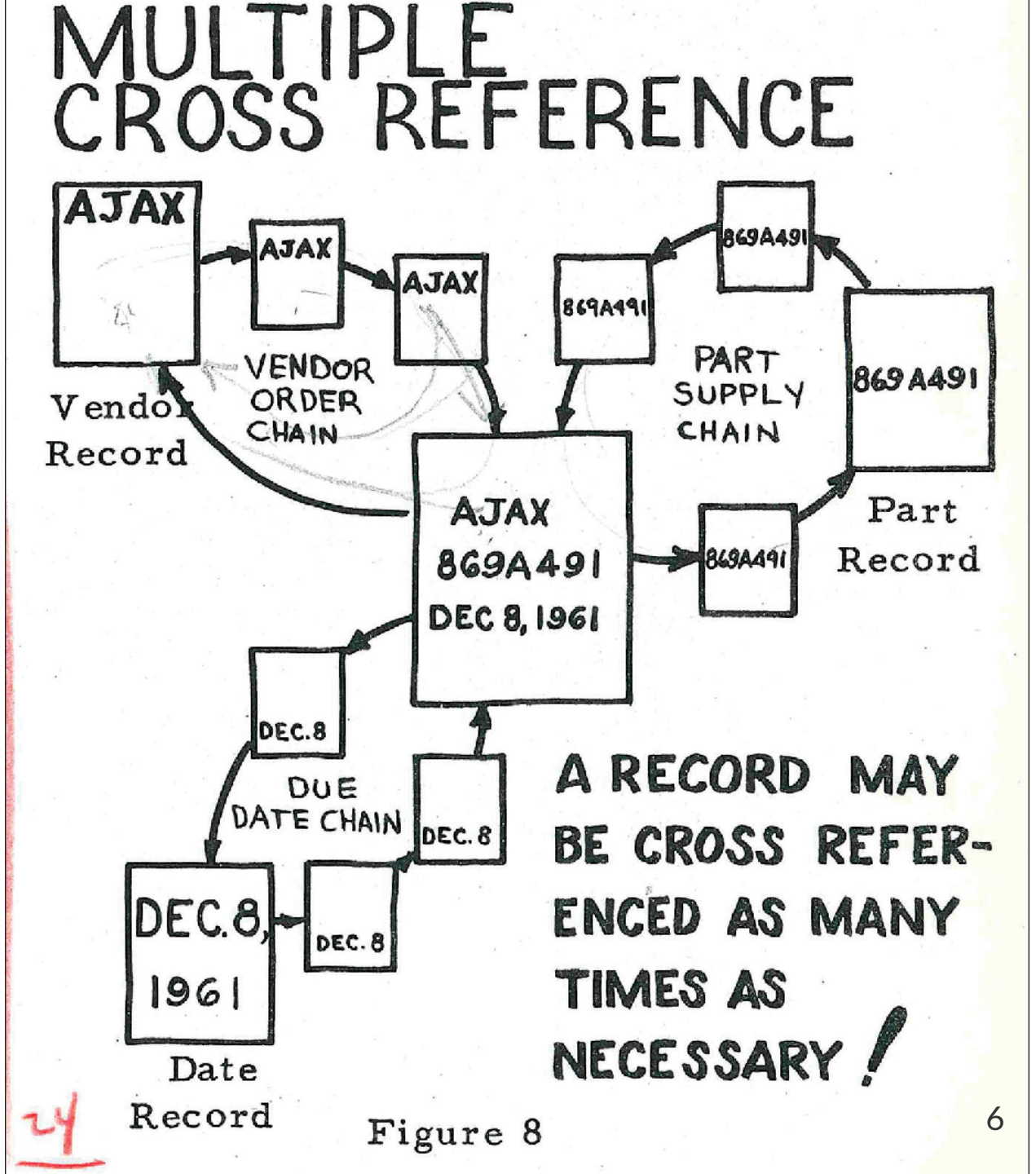
He received the **Turing Award** in 1973

Yes, GE was selling this in 1964 ⤵️



IDS

Network Database
(more on this later)



CODASYL

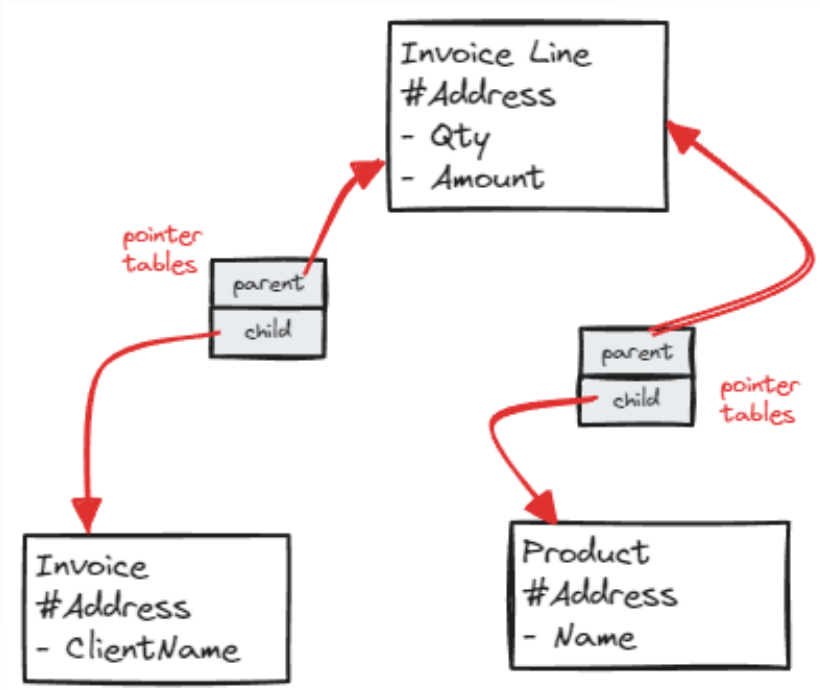
Conference **O**n **DA**ta **SY**stems **L**anguages

COBOL developers that wanted to come out with **APIs** for data storage and retrieval

Charles Bachman joins them in 1965, pushing many ideas from **IDS**

Network model

Codasyl and IDS were both using this model



Hierarchical model

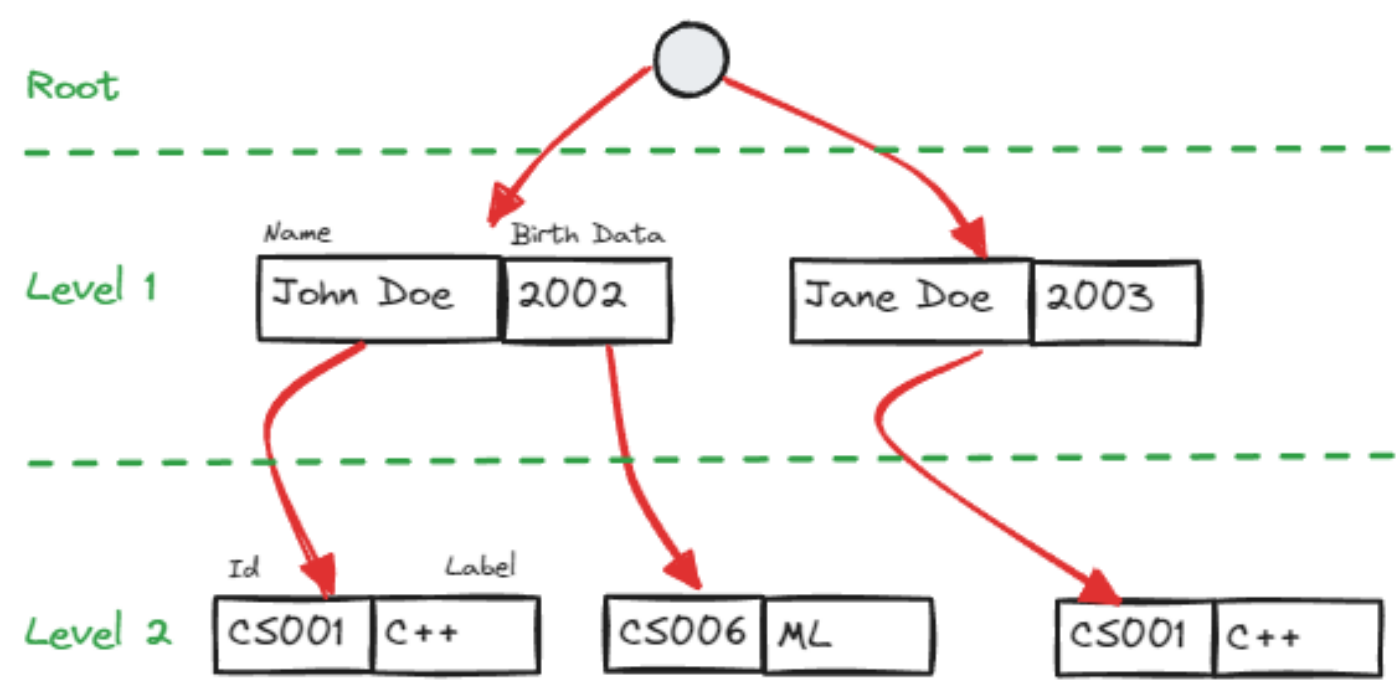
At the same time, another competing approach was the so-called **Hierarchical** Model

Pushed by IBM's **Information Management System** or IMS

Any idea why this was used? (tip on the right)

Hierarchical model

Think of an **xml**, **json** or **yaml** file

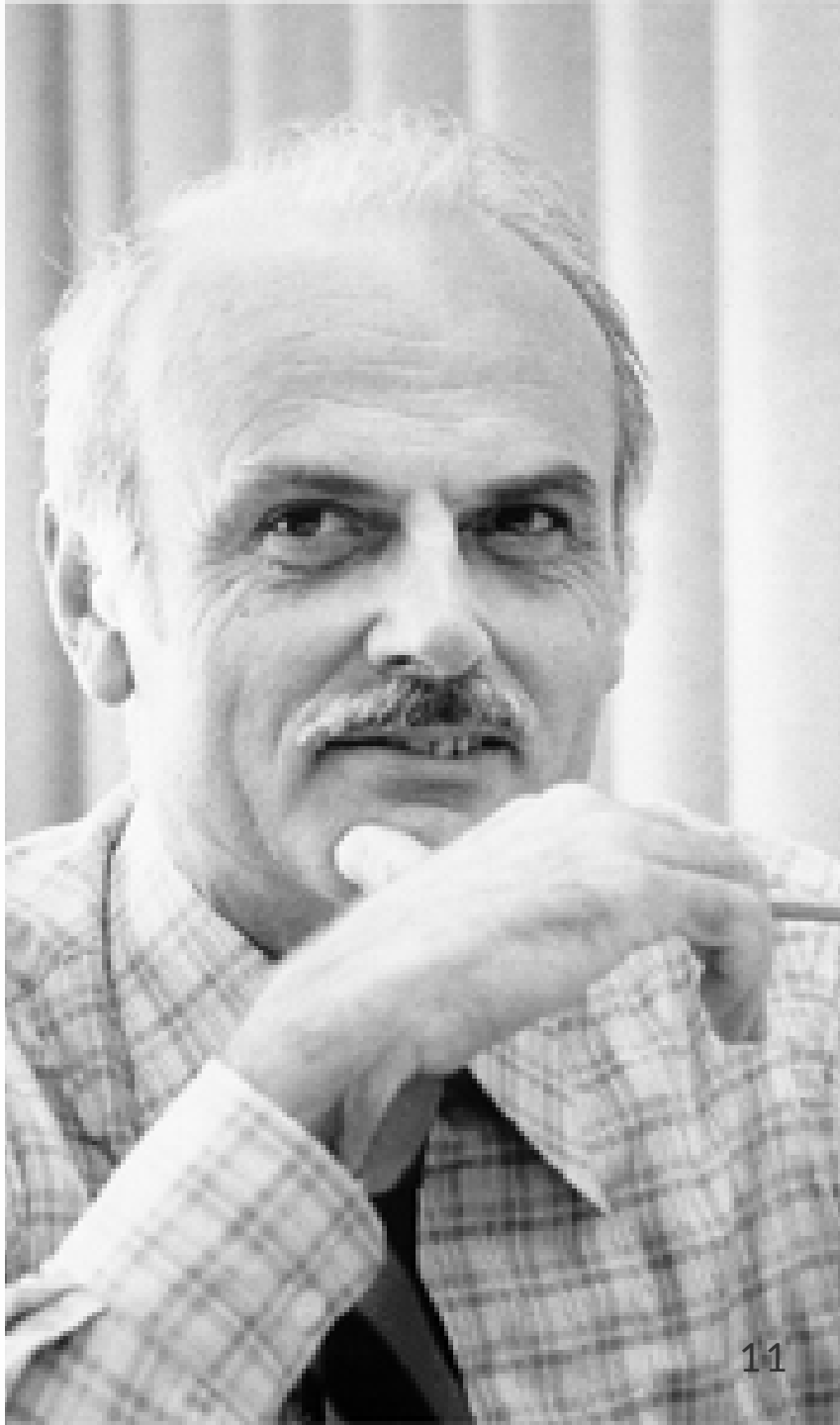


Edgar Codd

Mathematician, working at IBM

Had the idea of an abstraction over the way data is **physically** stored

The **Relational Model**

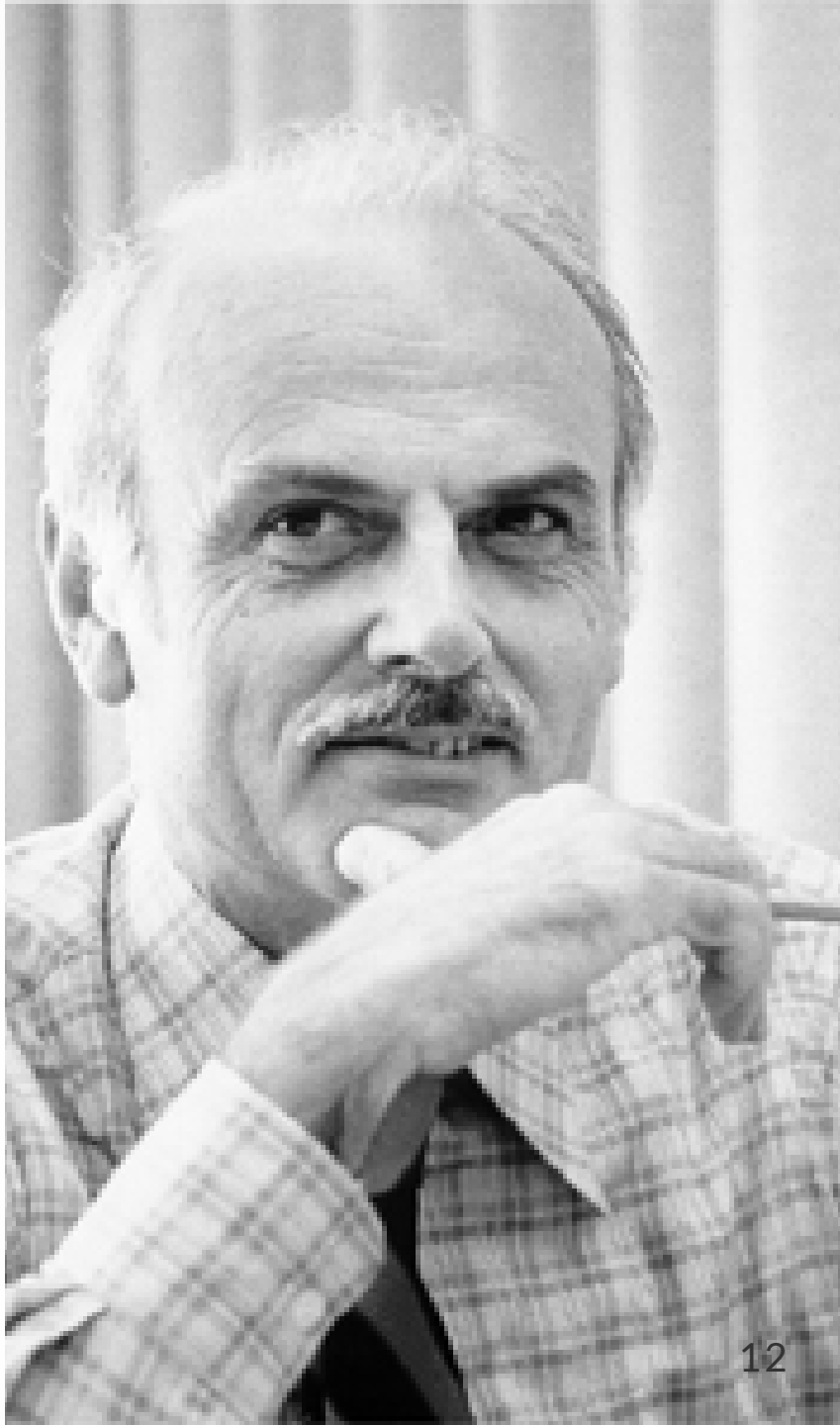


Edgar Codd

A Relational Model of Data for Large Shared Data Banks is published in 1970

One of the **most fundamental** papers in CS history

> 💡 I strongly advise you to read it



Moving to relational in the 70s

IBM develops **System R** following Edgar Codd's work

Michael Stonebraker starts **Ingres** at UC Berkeley and later quits to commercialize it

Larry Ellison founds **Oracle**

Market overtake in the 80s

IBM finally ships **DB2**

Oracle becomes the leader

Stonebraker comes back to Berkeley to start a new project named **Postgres**

A few events in the 90s

Microsoft enters the market following some high level of drama

PostgreSQL is Postgres with SQL support

MySQL is created

OLAP Cubes are the brand new hot thing, following Essbase

The "Internet Era" changes everything

Data volumes **explode** in the early 2000s

Big Tech giants start to look for other ways to manage data

BigTable

Jeff Dean from **Google** published *Bigtable: A Distributed Storage System for Structured Data* in 2006

Dynamo

Amazon published *Dynamo: Amazon's Highly Available Key-value Store* in 2007

It was meant to replace **Oracle** there

> About 70 percent of operations were of the key-value kind, where only a primary key was used and a single row would be returned. About 20 percent would return a set of rows, but still operate on only a single table.
>
> **Werner Vogels**

Dynamo-style databases

e.g. **Cassandra**

- All nodes are equal and there is no leader
- Consistent hashing is used to ensure distribution
- Data is replicated at least once

MapReduce

Jeff Dean - him again ! -published *MapReduce: Simplified Data Processing on Large Clusters* in 2004

MapReduce is a programming model using two functions

```
map(k1, v2) -> list(k2, v2)
reduce(k2, list(v2)) -> list(v2)
```

Rise (and fall) of Hadoop

Yahoo open sources its implementation of MapReduce.
All those projects become the Hadoop ecosystem.

Stonebraker disagreed

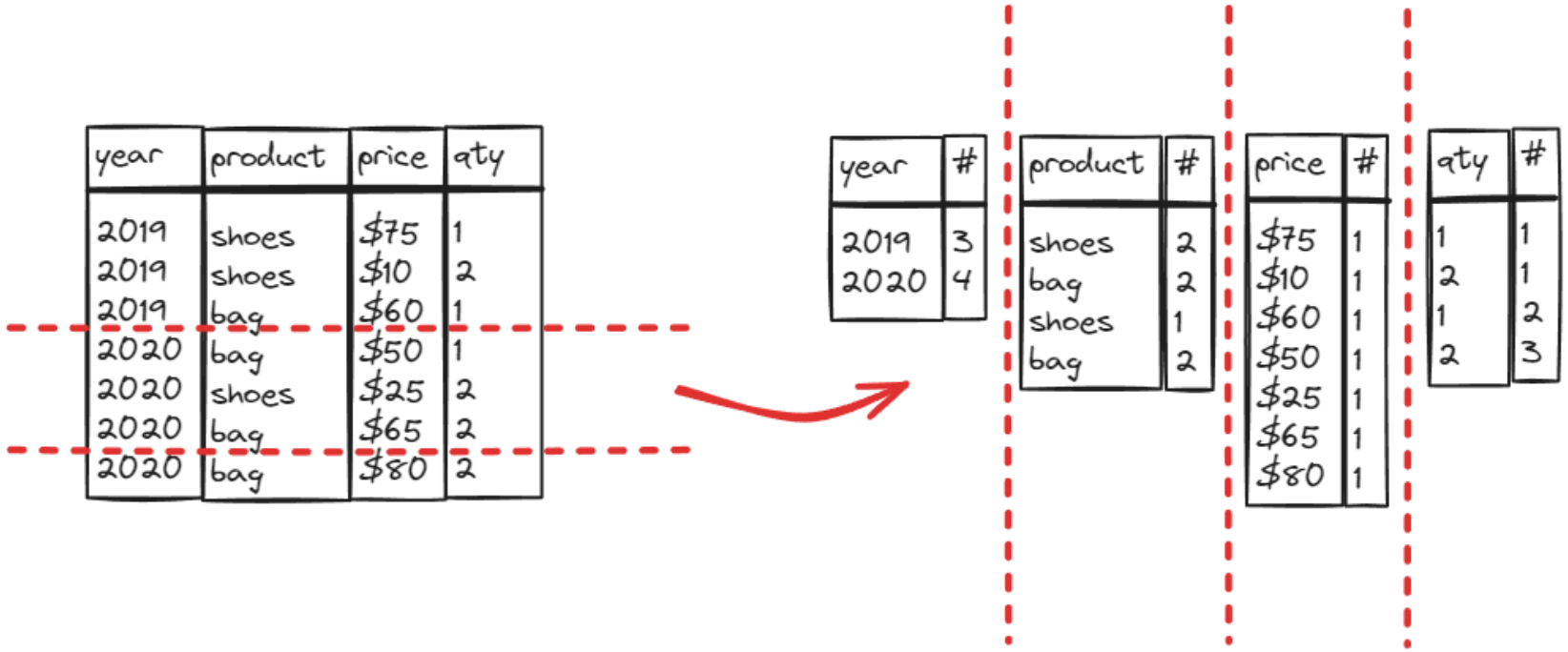
> 💬 *MapReduce represents a **giant step backwards**. The database community has learned the following three lessons since [...] 1968*
>
> 1. *Schemas are good*
> 2. *Separation of the schema from the application is good*
> 3. *High-level access languages are good*
>
> *MapReduce has learned **none** of these lessons and represents a **throw back to the 1960s***
>
> **Michael Stonebraker** - *MapReduce: A major step backwards* - 2008

CStore

In summary, there may be a substantial number of domain-specific database engines with differing capabilities off into the future. We are reminded of the curse "may you live in interesting times". We believe that the DBMS market is entering a period of very interesting times. There are a variety of existing and newly-emerging applications that can benefit from data management and processing principles and techniques. At the same time, these applications are very much different from business data processing and from each other — there seems to be no obvious way to support them with a single code line. The "one size fits all" theme is unlikely to successfully continue under these circumstances.

CStore

Storing in columns is **way** more efficient for analytic workloads



Massively Parallel Data Warehouses

Stonebraker - again - leaves research to start a database company named **Vertica**

Some others will implement the same idea (e.g. **Paraccel**)

Columnar Everywhere

Apache Parquet will be started by Twitter and Cloudera as a columnar filesystem for Hadoop clusters in 2013

It is still the most popular columnar file format, supported by pretty much anything

Distributed SQL

Jeff Dean - him again! - publishes *Spanner: Google's Globally-Distributed Database* - in 2012

- Multi-version
- Distributed
- Synchronously-Replicated

This led to multiple implementations

Cloud Native

Using object stores à la **S3**

Nodes as containers with attached disks for **caching**

Common Services - catalog and/or scheduler - hosted on **KV store**

Latest Trends

Specialized Engines

Graph databases for all use cases that require n-way joins, e.g. **Neo4J**

Embedded Column Stores such as **DuckDB**

ML engines such as **Chroma** to store Vectors / Embeddings

Relational databases aren't dead

Rank			DBMS	Database Model	Score		
Jan 2024	Dec 2023	Jan 2023			Jan 2024	Dec 2023	Jan 2023
1.	1.	1.	Oracle	Relational, Multi-model	1247.49	-9.92	+2.33
2.	2.	2.	MySQL	Relational, Multi-model	1123.46	-3.18	-88.50
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	876.60	-27.23	-42.79
4.	4.	4.	PostgreSQL	Relational, Multi-model	648.96	-1.94	+34.11
5.	5.	5.	MongoDB	Document, Multi-model	417.48	-1.67	-37.70
6.	6.	6.	Redis	Key-value, Multi-model	159.38	+1.03	-18.17
7.	7.	↑ 8.	Elasticsearch	Search engine, Multi-model	136.07	-1.68	-5.09
8.	8.	↓ 7.	IBM Db2	Relational, Multi-model	132.41	-2.19	-11.16
9.	↑ 10.	↑ 11.	Snowflake	Relational	125.92	+6.04	+8.66
10.	↓ 9.	↓ 9.	Microsoft Access	Relational	117.67	-4.08	-15.69

To be continued...