

The Rationale for Relational

Andrew Gregovic • Apr 26, 2018

Table of contents

History of DB models

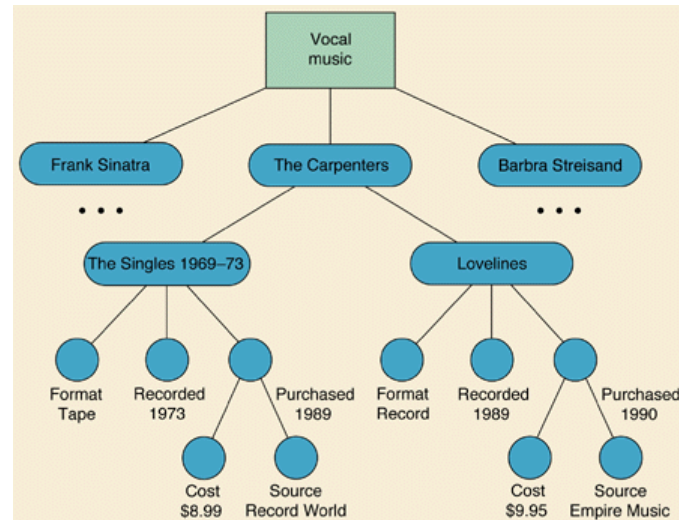
Relational vs SQL vs NoSQL vs
modern marketing

Why relational still matters

Further reading

OMG – Chocolate Fish!!!!

Hierarchical databases

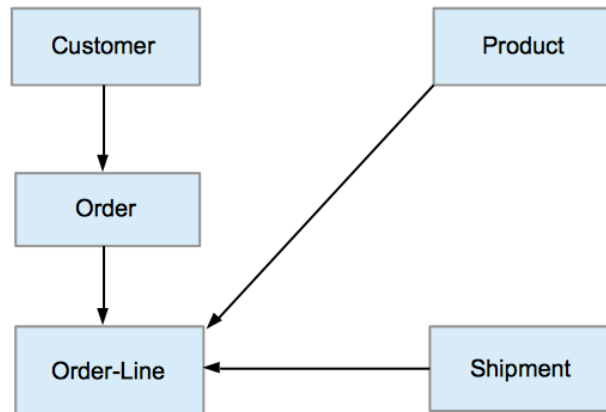


"A hierarchical database model is a data model in which the data is organized into a tree-like structure. The data is stored as records which are connected to one another through links."

"The hierarchical structure is used primarily today for storing geographic information and file systems."

Source: https://en.wikipedia.org/wiki/Hierarchical_database_model

Network databases



"Its distinguishing feature is that the schema, viewed as a graph in which object types are nodes and relationship types are arcs, is not restricted to being a hierarchy or lattice."

"Until the early 1980s the performance benefits of the low-level navigational interfaces offered by hierarchical and network databases were persuasive for many large-scale applications, but as hardware became faster, the extra productivity and flexibility of the relational model led to the gradual obsolescence of the network model in corporate enterprise usage."

Source: https://en.wikipedia.org/wiki/Network_model

Relational algebra and model

Relational algebra

- Set theory (union, intersect, minus...)
- Joins (Cartesian, natural, semi, outer, anti...)
- Aggregation

Source: https://en.wikipedia.org/wiki/Relational_algebra

Relational model

"The purpose of the relational model is to provide a ***declarative method*** for specifying data and queries: users directly state what information the database contains and what information they want from it, and ***let the database management system software take care*** of describing data structures for storing the data and retrieval procedures for answering queries."

Key formal modelling concepts: normal forms (e.g. 3NF), data integrity (PK constraints, FK constraints...)

Source: https://en.wikipedia.org/wiki/Relational_model

BORING!!!

WHY SHOULD I CARE?

WHERE'S THE CODEZ?

The 2 Golden Rules of Building Good Software

1. Don't Screw Up The Experience

2. Don't Screw up The Data

Quiz question:

What's wrong with
hierarchical data models?

Quiz question:

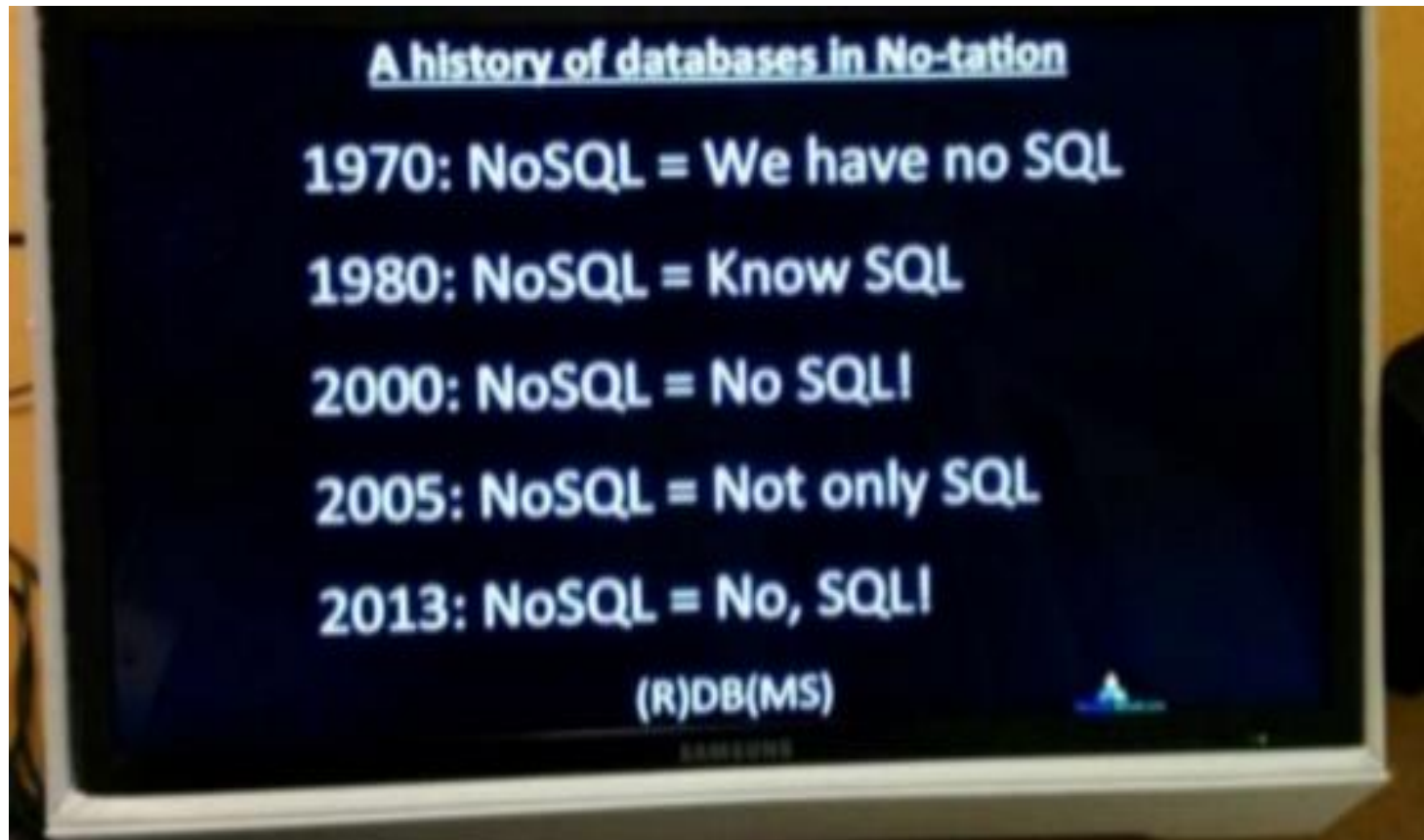
Is there such a thing as a
“schema-less database?”

NoSQL / Not Only SQL / Some QL data stores

Type	Examples
Key-Value Cache	Apache Ignite , Coherence , eXtreme Scale , Hazelcast , Infinispan , JBoss Cache, Memcached , Repcached, Velocity
Key-Value Store	ArangoDB , Flare, Keyspace, RAMCloud, SchemaFree, Aerospike , quasardb
Key-Value Store (Eventually-Consistent)	DovetailDB, Oracle NoSQL Database , Dynamo , Riak , Dynamite, Voldemort , SubRecord
Key-Value Store (Ordered)	Actord, FoundationDB , InfinityDB , Lightcloud, LMDB , Luxio, MemcacheDB , NMDB, TokyoTyrant
Data-Structures Server	Redis
Tuple Store	Apache River , Coord, GigaSpaces
Object Database	DB4O, Objectivity/DB , Perst , Shoal, ZopeDB
Document Store	ArangoDB , BaseX , Clusterpoint , Couchbase , CouchDB , DocumentDB , IBM Domino , MarkLogic , MongoDB , Qizx , RethinkDB
Wide Column Store	Amazon DynamoDB , Bigtable , Cassandra , Druid , HBase , Hypertable , KAI, KDI, OpenNeptune, Qbase

Source: <https://en.wikipedia.org/wiki/NoSQL>

SQL vs NoSQL



Credit: Mark Madsen

Building a NoSQL horse

NoSQL



<https://toggl.com/blog/build-horse-programming/>

Quiz question:

A document store relates to
which DB model?

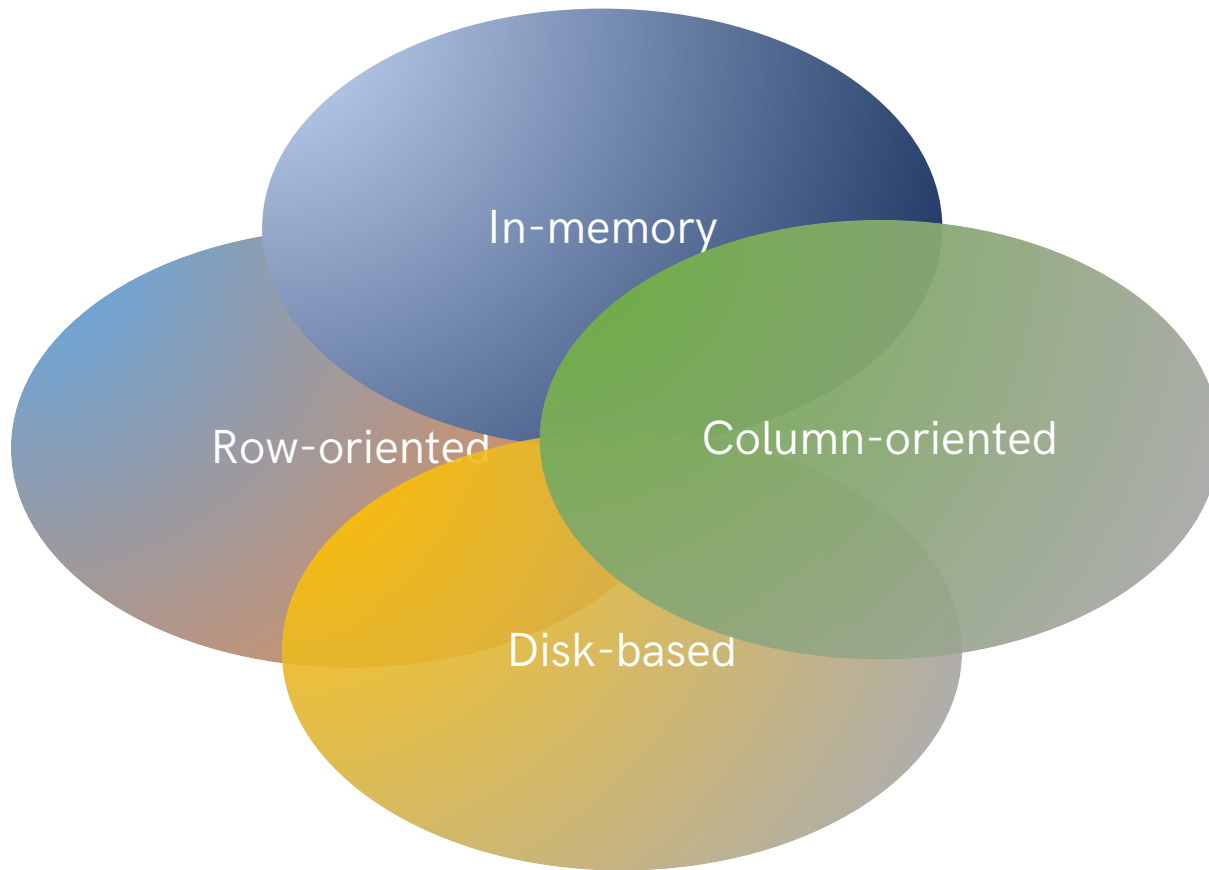
Quiz question:

A graph DB relates to which
DB model?

The 2 Big + 1 Not-so-big Misconceptions:

1. Relational = traditional row-based DB
2. Relational = not scalable or highly available
3. SQL = Relational

Not all relational databases are created equal



Typical claim #1

Flexible schemas = good

Rigid schemas = bad

Typical claim #2

SQL/relational DBs
cannot be clustered

Typical claim #3

SQL/Relational databases
can't do efficient graph
queries

NewSQL kids on the block



 Google Cloud Platform

cloud.google.com/spanner

splice
MACHINE

VOLTDDB

APACHE
PHOENIX

CockroachDB 1.0


NUODB®

DO YOUR DUE DILIGENCE!

Key Benefits of Relational DBs

1. ACID
2. Abstraction
3. Query Optimization
4. Portability

Further reading

When not to use NoSQL

<http://www.sarahmei.com/blog/2013/11/11/why-you-should-never-use-mongodb/>

Deeper dive into isolation models, CAP and 25 questions when choosing a DB

<https://medium.com/@andrew.gregovic/think-twice-before-dropping-acid-and-throw-your-cap-away-dbe0d6171dc0>

GREAT INDIAN **DEVELOPER**TM **SUMMIT** 2019

Conference : April 23-26, Bangalore



Register early and get the best discounts!



www.developersummit.com



[@greatindiandev](https://twitter.com/greatindiandev)



bit.ly/gidslinkedin



facebook.com/gids19



bit.ly/saltmarchyoutube



flickr.com/photos/saltmarch/