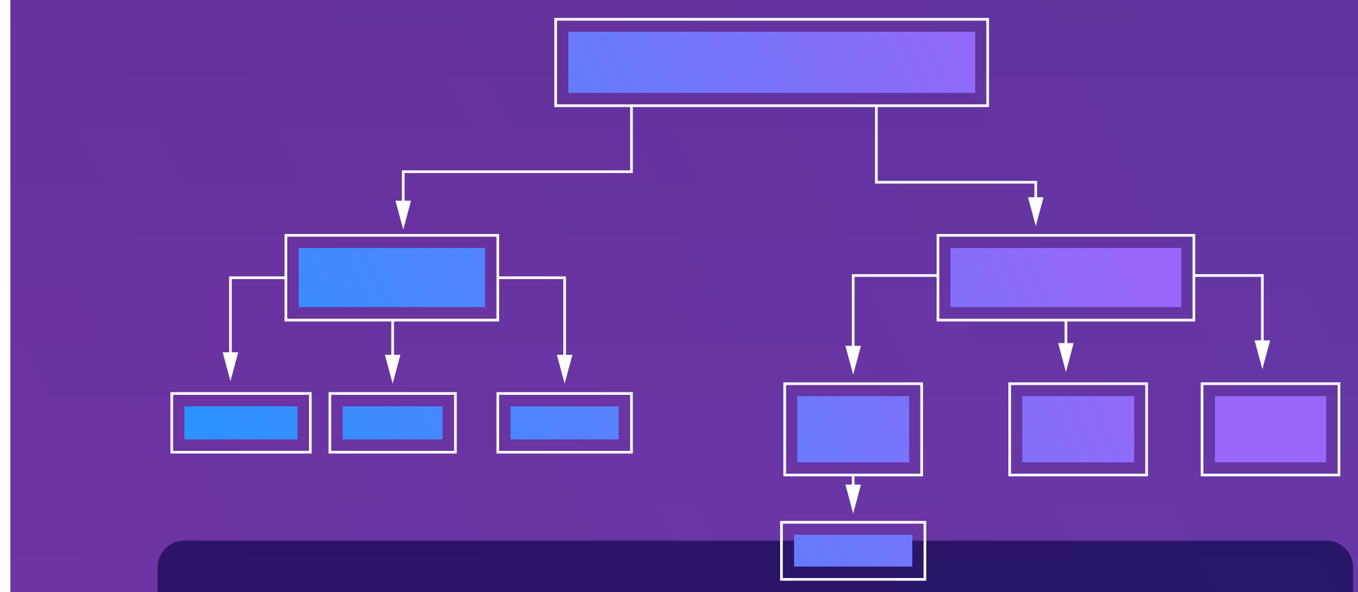
WHY ANOTHER DATABASE?

The Evolution of the Transactional Database

A database is a collection of information that is organized so it can be easily accessed, managed, and updated.



IMS (INFORMATION MANAGEMENT SYSTEM) FROM IBM

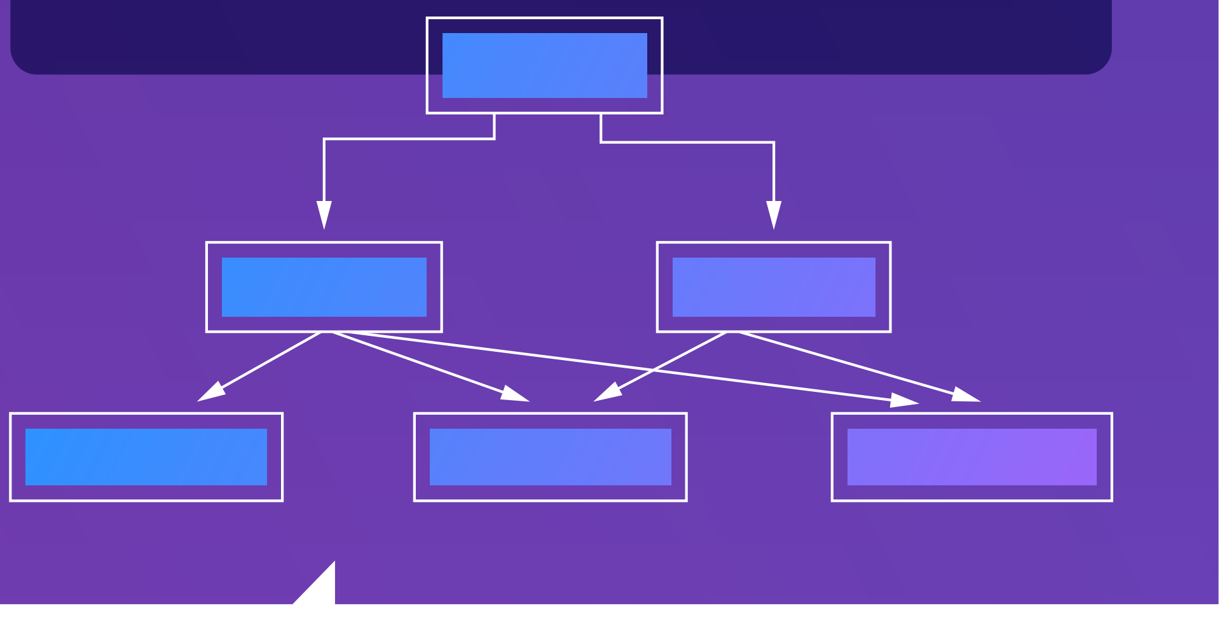
- Hierarchical data model
- Data organized into tree-like structure with one-to-many relationships between 2 different types of data

1966

1969

NETWORK DATA MODEL

- Extension of hierarchical data model
- Support many-to-many (m-n) relations



INGRES

First practical and efficient implementation of a relational model (RM) which represents a database as a collection of relations.

1973

ORACLE

First commercial SQL-based RDBMS

1979

SQL

SQL becomes standard language for databases, after being developed at IBM in the early 1970s

1980'S

GREAT ON A SINGLE MACHINE BUT STRUGGLES TO SCALE

OTHER RELATIONAL DATABASES APPEAR:

1983- **IBM DB2**
1987 – **Sybase**
1989 – **Postgres**
1989 – **Microsoft SQL Server**
1995 – **MySQL**

2000'S

EMERGENCE OF NoSQL DATABASE

Focused on horizontal scalability without having to compromise on performance as monolithic SQL database.

MongoDB –2007
Cassandra, HBase – 2008
Couchbase – 2011
AWS DynamoDB – 2012
Azure CosmoDB – 2017

NoSQL stood for "no support for SQL" but changed to "not only SQL" as it became clear that it must coexist with—not replace— SQL.

AND NewSQL DATABASE

NewSQL looked to provide the scalability of NoSQL systems for online transactional processing (OLTP) workloads while maintaining the ACID guarantees of a traditional database system.

ClustrixDB –2006
Vitess, VoltDB – 2010
Citus, NuoDB - 2011

THE PATH TO DISTRIBUTED SQL

Modern applications need modern databases

2012

GOOGLE SPANNER

Entirely new SQL database that can horizontally scale, seamlessly geo-replicate, and easily tolerate infrastructure failures

DISTRIBUTED SQL DATABASE

Distributed SQL seeks to provide the scalability of NoSQL systems for online transactional processing (OLTP) workloads while maintaining the ACID guarantees of a traditional database system.

MariaDB - 2010
Amazon Aurora - 2014
TIDB - 2015
CockroachDB – 2015
YugabyteDB - 2016

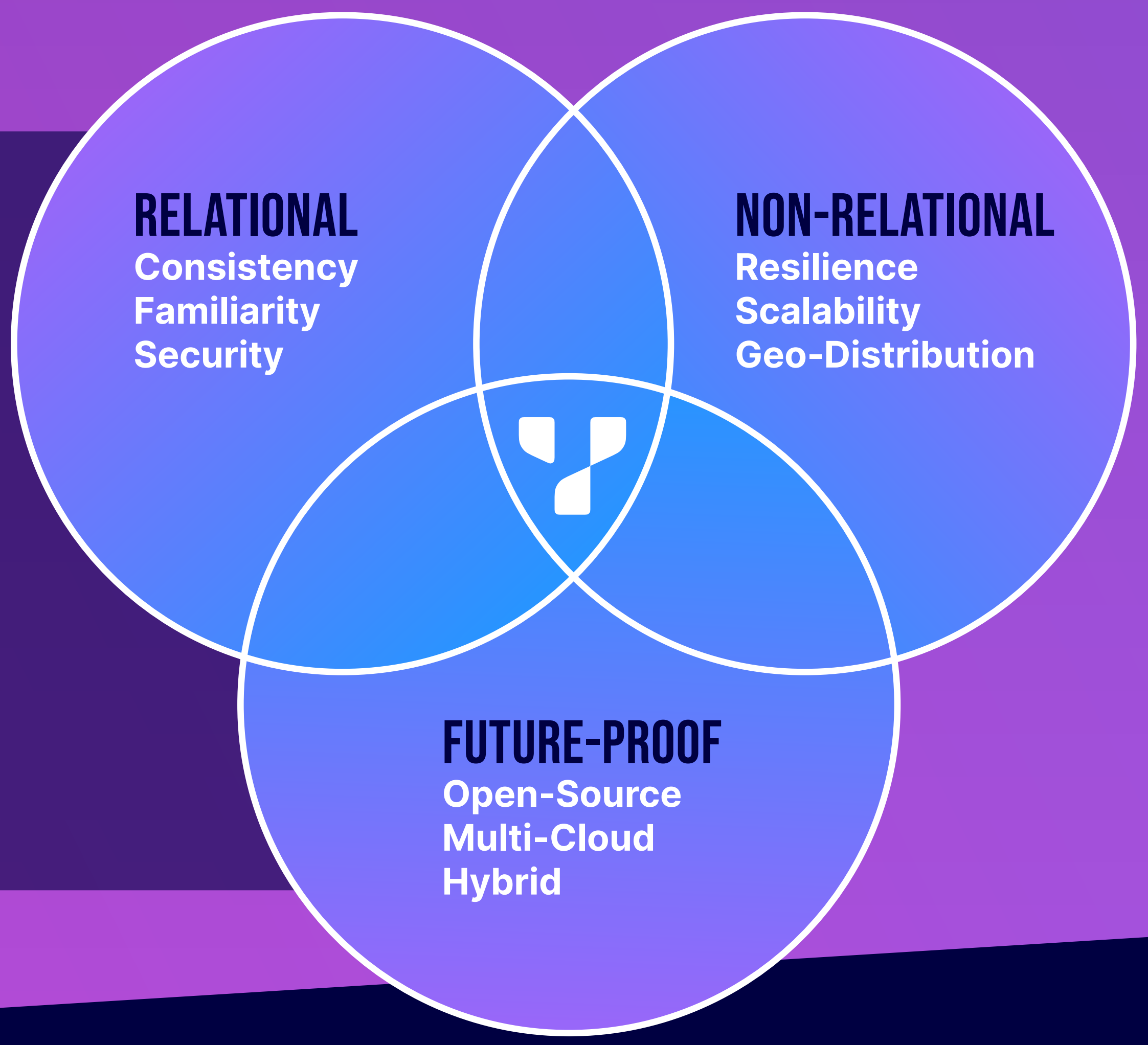
2014

2016+

THE FUTURE OF DISTRIBUTED SQL—YUGABYTEDB

COMBINES THE BEST OF SQL AND NOSQL

- Inspired by Google Spanner
- Open source
- Strong transactional consistency
- Run anywhere (multi-cloud)
- Geo-distribution
- PostgreSQL-compatible
- Database-as-a-service option offered



RELATIONAL
Consistency
Familiarity
Security

NON-RELATIONAL
Resilience
Scalability
Geo-Distribution

FUTURE-PROOF
Open-Source
Multi-Cloud
Hybrid

yugabyteDB

yugabyte.com | contact@yugabyte.com

Learn more:
Evolution of Transactional Databases