GOOGLE CLOUD PLATFORM

CLOUD SPANNER

WHAT IS CLOUD SPANNER?

Description

- Mission-critical, relational database service with transactional consistency, global scale and high availability.

Good for

- Mission-critical applications
- High transactions
- Scale + Consistency requirements

Common Workloads

- Adtech
- Financial services
- Global supply chain
- Retail

FEATURES

- Fully managed database service with a global scale
- Traditional relational semantics: schemas, ACID transactions, SQL
- Automate, synchronous replication within/across regions for availability, maintained by site reliability engineers
- Uses nodes for scalability - each node provides up to 2 TiB of storage
- No backup solution available
- Used at Google for Google Play, Adwords, etc

TRADITIONAL VS. SPANNER

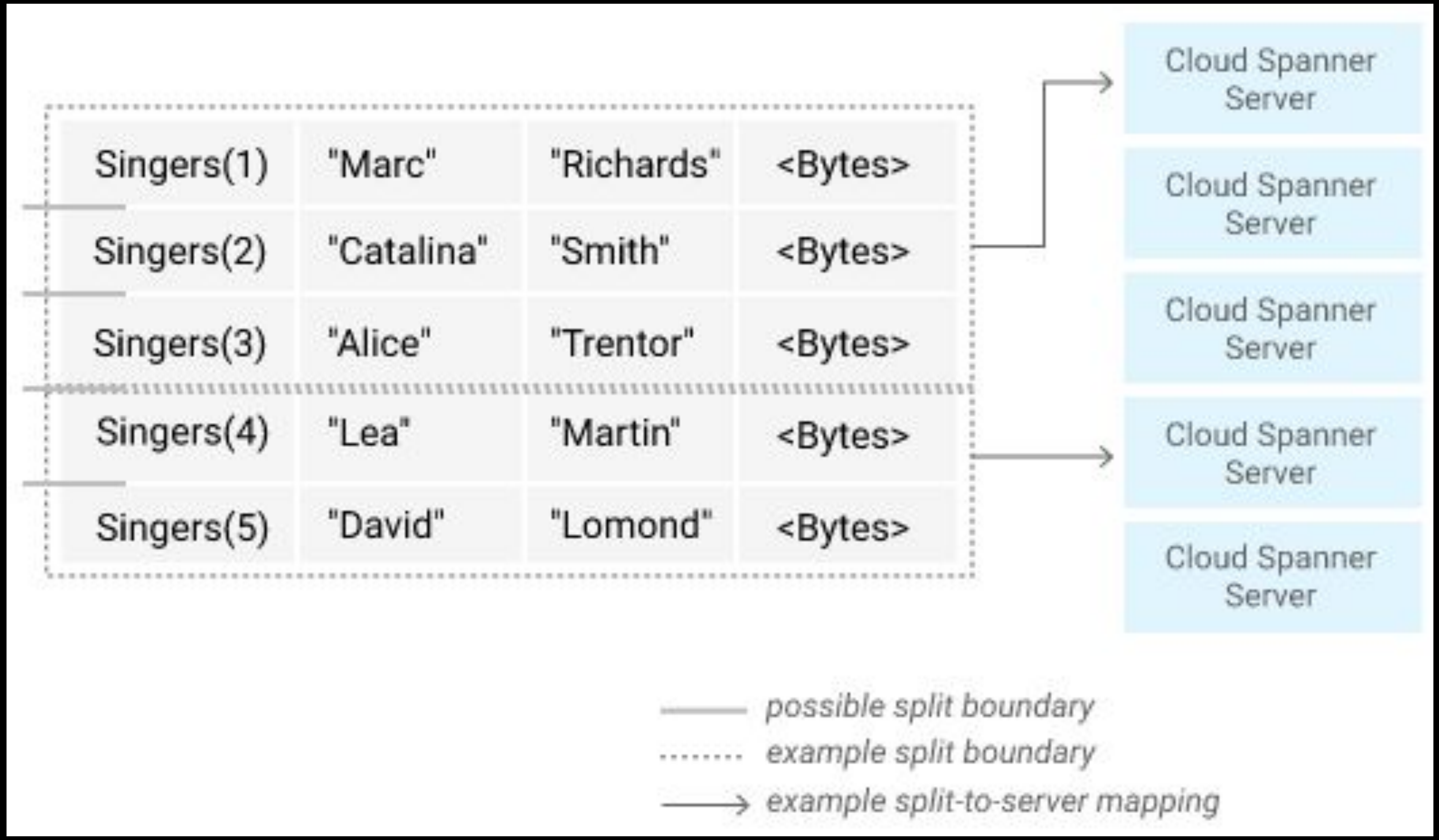
	SPANNER	TRADITIONAL RELATIONAL	TRADITIONAL NON-RELATIONAL
Schema	Yes	Yes	No
SQL	Yes	Yes	No
Consistency	Strong	Strong	Eventual
Availability	High	Failover	High
Scalability	Horizontal	Vertical	Horizontal
Replication	Automatic	Configurable	Configurable

SCHEMA AND DATA MODEL

- Tables look like relational database tables in that they are structured with rows, columns, and values, and they contain primary keys.

- Data is strongly typed: you must define a schema for each database and that schema must specify the data types of each column of each table

- Cloud Spanner divides your data into chunks called "splits” which can move to different nodes/servers



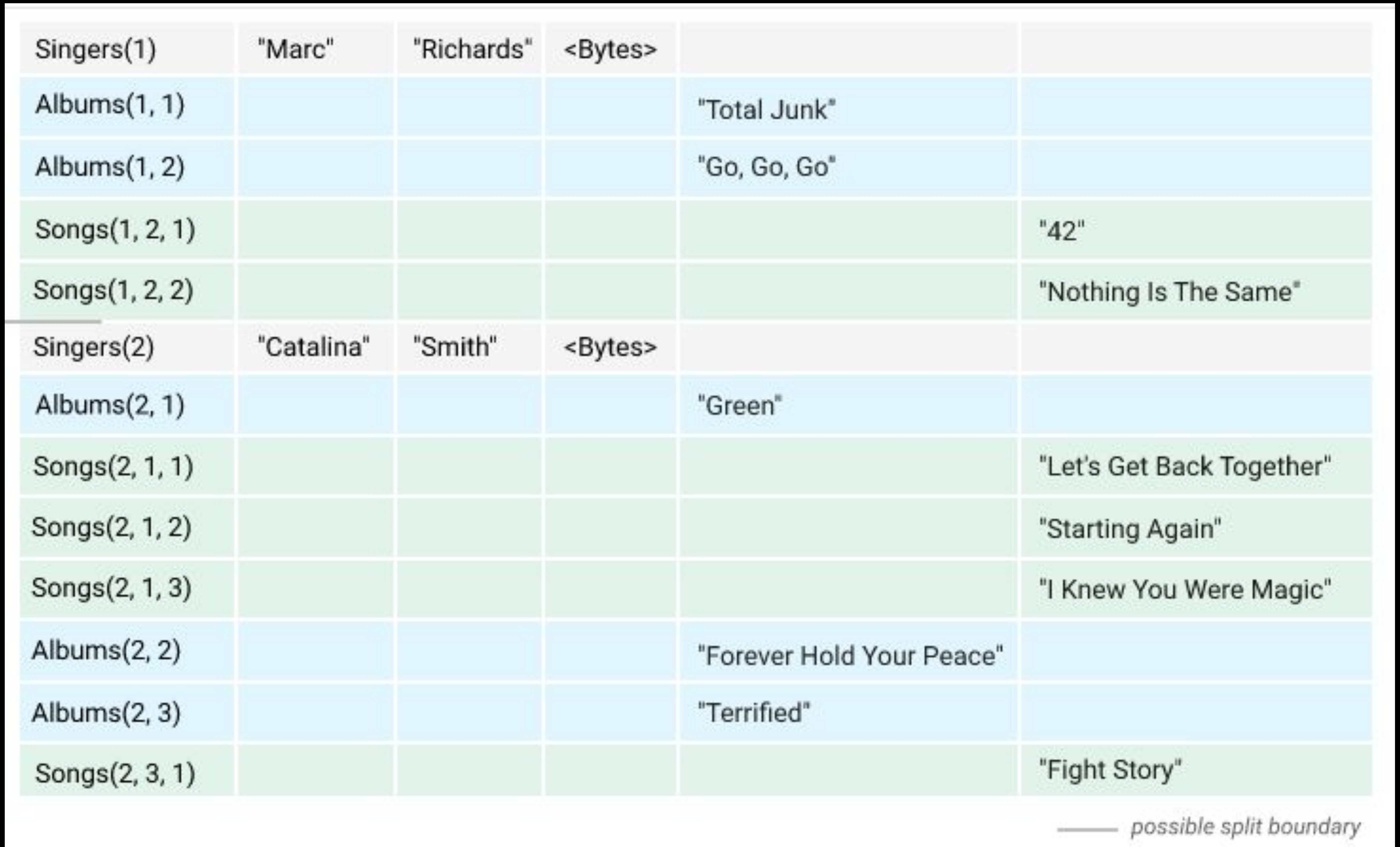
Singers(1)	"Marc"	"Richards"	<Bytes>
Singers(2)	"Catalina"	"Smith"	<Bytes>
Singers(3)	"Alice"	"Trentor"	<Bytes>
Singers(4)	"Lea"	"Martin"	<Bytes>
Singers(5)	"David"	"Lomond"	<Bytes>

INTERLEAVE DATA

- An **interleaved table** is a table that you declare to be a child of another table because you want the rows of the child table to be physically stored together with the associated parent row.

- E.g. Three tables – Singers, Albums and Songs

- Physically stored as one table, but gets split across nodes



Singers(1)	"Marc"	"Richards"	<Bytes>		
Albums(1, 1)				"Total Junk"	
Albums(1, 2)				"Go, Go, Go"	
Songs(1, 2, 1)					"42"
Songs(1, 2, 2)					"Nothing Is The Same"
Singers(2)	"Catalina"	"Smith"	<Bytes>		
Albums(2, 1)				"Green"	
Songs(2, 1, 1)					"Let's Get Back Together"
Songs(2, 1, 2)					"Starting Again"
Songs(2, 1, 3)					"I Knew You Were Magic"
Albums(2, 2)				"Forever Hold Your Peace"	
Albums(2, 3)				"Terrified"	
Songs(2, 3, 1)					"Fight Story"

possible split boundary

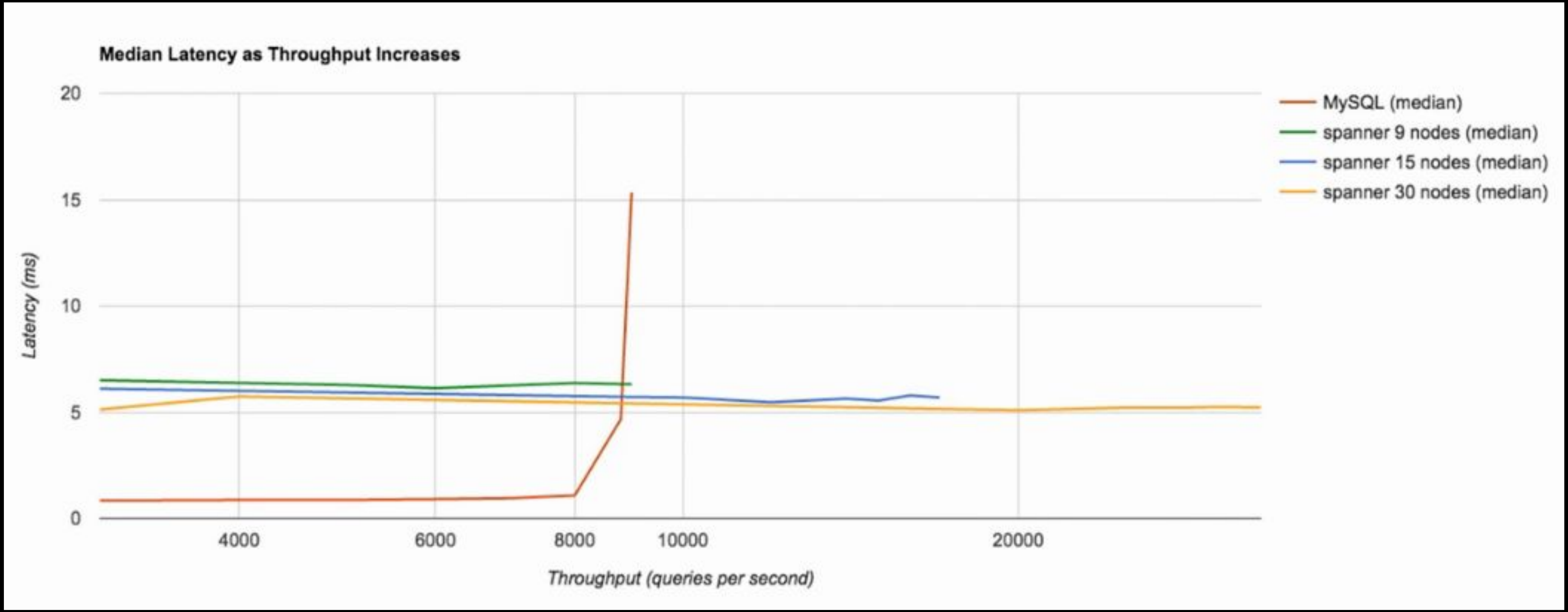
REPLICATION

- Cloud Spanner creates replicas of each database split
- Three types: Read-write, Read-only, Witness
- Certain parts of the data in the replicas is owned by different nodes – custom algorithm automatically manages data
- Benefits includes Data availability, Geographic locality, Single database experience and Easier application development

DEMO: CREATE CLOUD SPANNER

1. Select/create project
2. Navigate to the Cloud Spanner dashboard
3. Create Instance
4. Create a new database
5. Create two interleaved tables

HORIZONTAL SCALABILITY



- **Increase the number of nodes and splits are automatically optimized across replicas over regions (Quizelet app) – 700 GB, 6 Billion rows**

TRUETIME & EXTERNAL CONSISTENCY

- TrueTime is a distributed clock that enables applications to generate monotonically increasing timestamps
- Cloud spanner uses this to timestamp transactions, allowing it to perform consistent reads across an entire database and across multiple Cloud regions without blocking writes.
- With External Consistency, the system behaves as if all transactions were executed sequentially, even though Cloud Spanner actually runs them across multiple servers
- Extrernal consistency is a stronger property than *strong consistency* as it doesn't block data during *strong reads*