Return to Table of Contents

Choose a Lesson

Cloud Spanner Overview

Data Organization and Schema

Cloud Spanner Overview

a "kind" of SQL

Next

What is Cloud Spanner?

- Fully managed, highly scalable/available, relational database
- Similar architecture to Bigtable
- "NewSQL" is used for strong transactional consistency or ACID compliance

What is it used for?

- Mission critical, relational databases that need strong transactional consistency (ACID compliance)
- Wide scale availability
- Higher workloads than Cloud SQL can support
- Standard SQL format (ANSI 2011)

Horizontal vs. vertical scaling

- Vertical = more compute (resource) on single instance (CPU/RAM)
- Horizontal = more instances (nodes) sharing the load

Compared to Cloud SQL

- Cloud SQL = Cloud incarnation of *on-premises* MySQL database
- Spanner = designed from the ground up for the cloud
- Spanner is not a 'drop in' replacement for MySQL
  - Not MySQL/PostreSQL compatible, so not a lift-and-shift version of SQL
  - Work required to migrate
  - However, when making transition, don't need to choose between consistency and scalability

Return to Table of Contents

Choose a Lesson

Cloud Spanner Overview

Data Organization and Schema

Cloud Spanner Overview

Previous

Next

trade-off

Transactional Consistency vs. Scalability
Why not both?

	Cloud 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

Primary purpose of Cloud Spanner:
No compromises relational database

best for both ∞ database

Return to Table of Contents

Choose a Lesson

Cloud Spanner Overview

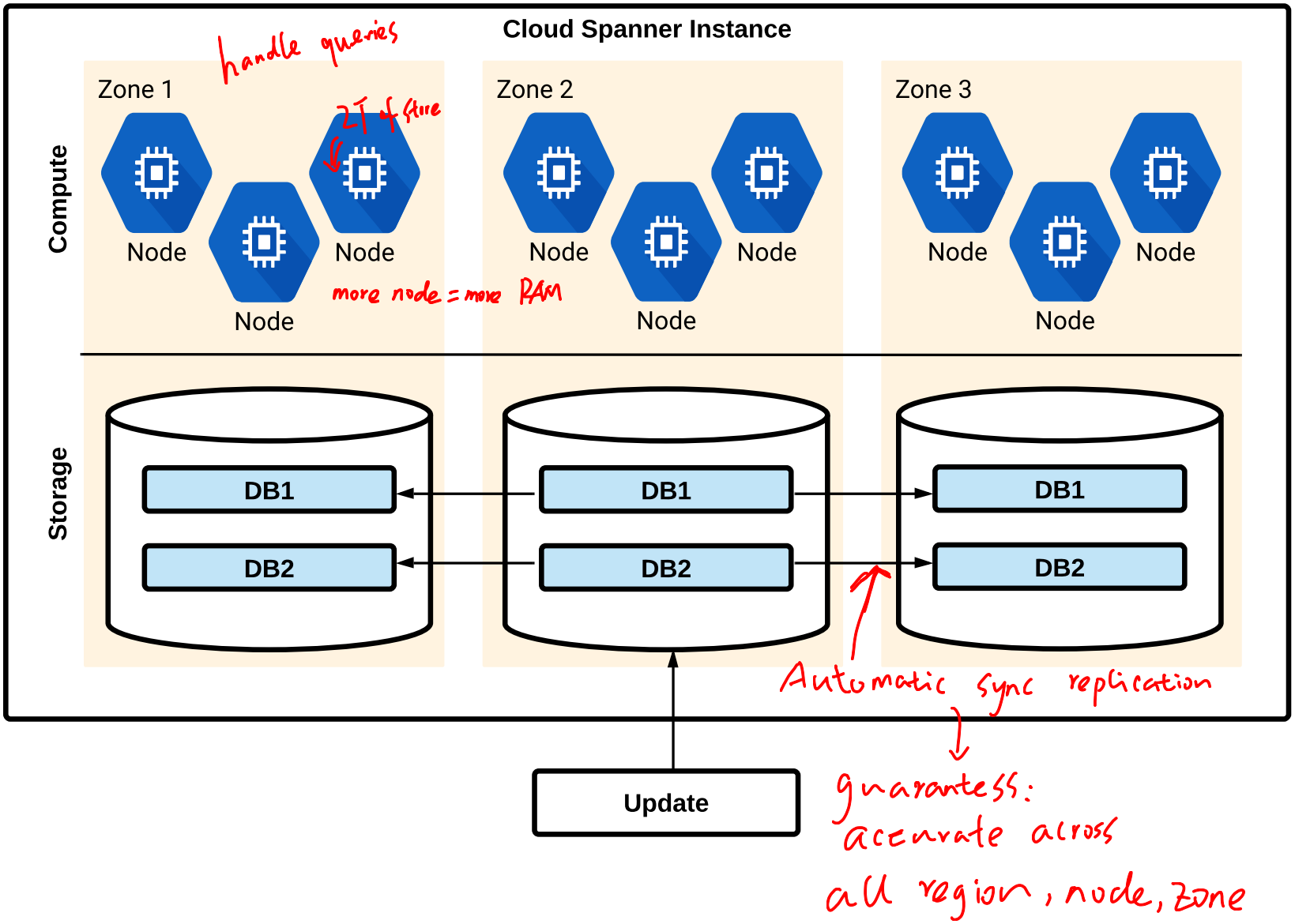
Data Organization and Schema

Cloud Spanner Overview

Previous

Next

Cloud Spanner Architecture
(similar to Bigtable)



Return to Table of Contents

Choose a Lesson

Cloud Spanner Overview

Data Organization and Schema

Cloud Spanner Overview

Previous

Identity and Access Management (IAM)

- Project, Instance, or Database level
- roles/spanner. (role name)
  1. Admin - Full access to all Spanner resources
  2. Database Admin - Create/edit/delete databases, grant access to databases
  3. Database Reader - read/execute database/schema
  4. Viewer - view instances and databases
     - Cannot modify or read from database

Roles

* IAM Questions will be on the exam for any given services

The Data Dossier

Return to Table of Contents

Choose a Lesson

Data Organization and Schema

Organization

Next

- RDBMS = tables
- Supports SQL joins, queries, etc
- Same SQL dialect as BigQuery
- Tables are handled differently (compared to other SQL databases)
  - Parent/child tables
  - Interleave Data Layout

Typical Relational Database

Two sets of related data = Two tables

SingerId	SingerName
1	Beatles
2	U2
3	Pink Floyd

SingerId	AlbumId	AlbumName
1	1	Help!
1	2	Abbey Road
3	1	The Wall

Spanner Interleave Tables for 3 Tables

parent → child → child of child

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"

Return to Table of Contents

Choose a Lesson

Cloud Spanner Overview

Data Organization and Schema

Data Organization and Schema

Previous

Primary keys and Schema

- How to tell which child tables to store with which parent tables
- Usually a natural fit
  - 'Customer ID'
  - 'Invoice ID' (child table)
- Avoid hotspotting – like bigtable
  - No sequential numbers for primary key
  - No timestamps (also sequential)
    - Use descending order if timestamps required

2 Recommends