



Google Cloud Spanner Use Case

Use Case: Managing Global Databases with Google Cloud Spanner

GENERAL CHARACTERISTICS

Intent	To manage and scale globally distributed relational databases using Google Cloud Spanner.
Scope	Implementation of a globally consistent database solution for mission-critical applications.
Level	System-level.
Client	Confidential (Multinational Banking Corporation).
Last Update	03/12/2024
Status	Finalized.
Stage	Deployment and Optimization.

ACTORS

Primary Actor	Database Administrator.
Secondary Actors	Application Developers, IT Operations Team, Compliance Officers.

PREREQUISITES

Static Preconditions	<ul style="list-style-type: none">- Google Cloud Project set up with Cloud Spanner API enabled.- Database schema designed and optimized for scalability.
Dynamic Preconditions	<ul style="list-style-type: none">- Data migration strategy planned and validated.- IAM roles and permissions configured for secure database access.
Assumptions	<ul style="list-style-type: none">- Client requires globally consistent data with high availability.- Transactions must meet stringent latency and reliability standards.

TRIGGERS

Trigger Event	The client needed a globally distributed database to handle real-time transactional data with high availability and consistency.
---------------	--

EXPECTED OUTCOME

Success Postcondition	<ul style="list-style-type: none">- Database supports global transactions with strong consistency.- Performance meets SLAs for latency and
-----------------------	---





Google Cloud Spanner Use Case

	availability.
Failed Postcondition	- Transaction delays or inconsistencies impact mission-critical applications.

OPERATIONS AND CONCEPTS

Operations	<ol style="list-style-type: none"> 1. Designed a schema optimized for globally distributed transactions. 2. Migrated existing relational data to Google Cloud Spanner. 3. Configured replication policies to ensure high availability. 4. Implemented IAM policies to enforce secure access control. 5. Monitored database performance using Cloud Monitoring and Logs Explorer. 6. Regularly reviewed and optimized queries for latency and cost efficiency.
Concepts	<ul style="list-style-type: none"> - Cloud Spanner: A globally distributed, strongly consistent relational database. - Schema Design: Optimizes performance for distributed workloads. - Replication: Ensures high availability across multiple regions.

MAIN SUCCESS SCENARIO

Step 1	Analyzed the client's requirements for global consistency and high availability.
Step 2	Designed a schema to support distributed transactions efficiently.
Step 3	Migrated the client's relational data to Google Cloud Spanner.
Step 4	Configured replication policies to ensure global availability.
Step 5	Implemented secure IAM roles and permissions for database access.
Step 6	Monitored performance and optimized queries for cost and latency.
Step 7	Delivered a robust database solution meeting SLA requirements.

