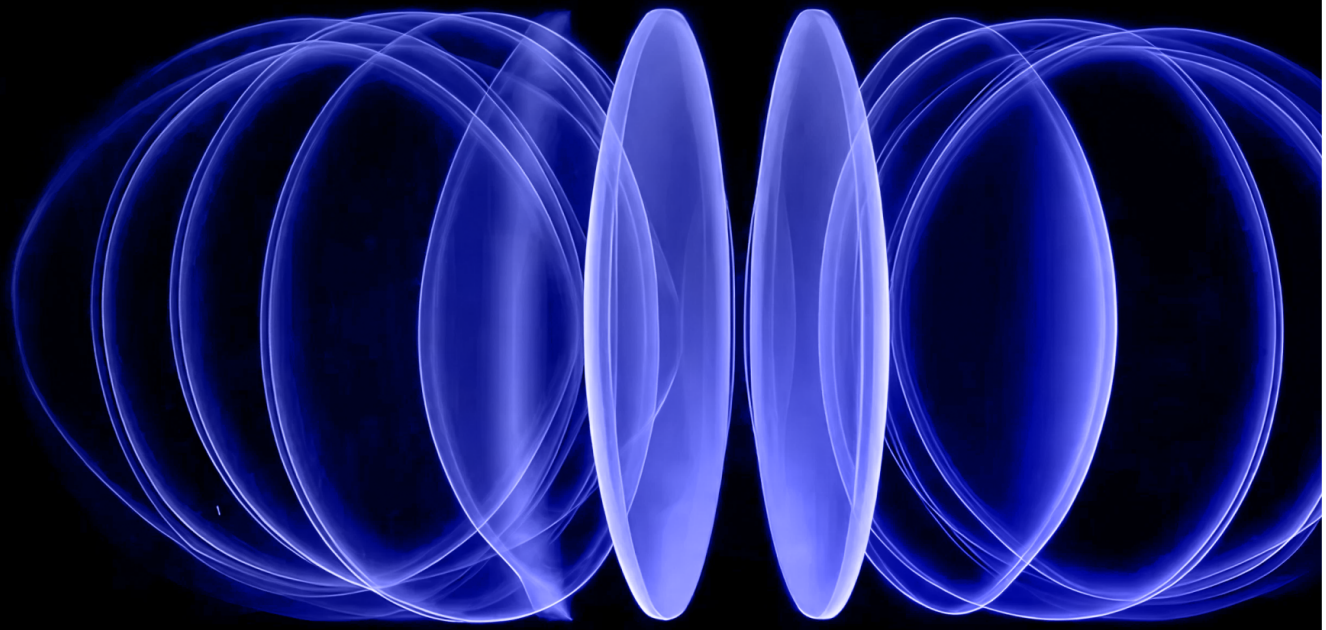
E-book

HOW ORGANIZATIONS ARE MODERNIZING DATA IN GOOGLE CLOUD

How three organizations handled stored procedures



softserve | Google Cloud

When companies move to the cloud, the assumption is often that the hardest part is moving the data. It isn't. The real challenge is when teams run into logic that's been living inside their databases for years in the form of stored procedures.

Platforms like Oracle, SQL Server, and DB2 encouraged teams to embed business rules directly in the database. Over time, stored procedures took on far more responsibility than originally intended. They calculate prices, enforce rules, generate reports, process batches, and quietly keep critical workflows running, often with little documentation and limited visibility.

This becomes a serious obstacle during modernization. The data is only part of the equation. The business logic embedded in stored procedures is just as critical, and in many cases, more difficult to transition.

That legacy logic does not map cleanly to modern cloud architectures. It complicates adoption of cloud-native patterns, horizontal scalability, CI/CD, modern analytics tooling, and managed services. As a result, stored procedures must be rewritten, re-implemented, or re-architected to align with how today's cloud platforms are designed to operate.

Drawing from three real Google Cloud migrations delivered by SoftServe, this paper examines how organizations in healthcare, insurance, and financial software handled stored procedures while modernizing their data platforms.

Each story centers on a different Google Cloud destination — AlloyDB, BigQuery, and Cloud Spanner — and each required a fundamentally different approach to preserving business logic.

Google Cloud Platforms

BigQuery

A serverless data warehouse designed for large-scale analytics

AlloyDB

A high-performance, PostgreSQL-compatible database for transactional and analytical jobs

Cloud Spanner

A globally distributed database built for applications requiring massive scale and strong consistency

What Stored Procedures Really Are (and Why They Matter)

In legacy environments like Oracle, SQL Server, and DB2, stored procedures became a convenient way to move fast. Instead of writing logic in application code, teams put it directly in the database, close to the data, and easy to execute. Over time, what started as a shortcut became the standard way systems were built.

The problem is that this logic grows organically. New rules get added. Edge cases get patched in. Procedures start calling other procedures. Before long, a significant portion of the business's behavior lived inside the database.

From a business standpoint, they represent institutional knowledge. They encode how the company operates. If that logic changes, even slightly, the impact shows up immediately in broken workflows, incorrect reports, or compliance issues.

From a technical standpoint, the risk is just as real. Stored procedures are tightly tied to the databases they were written for, relying on specific languages, execution models, and performance assumptions. Even when a new platform supports stored procedures, the differences are often large enough that direct migration isn't realistic.

That's why stored procedures matter so much during Google Cloud migrations.

Google Cloud Changes the Question Entirely

In legacy systems, the database often was the center of the architecture. Logic, data, and processing all lived together. Google Cloud changes that model. Its platforms are designed around specialization: analytics engines do analytics, globally distributed databases focus on scale and consistency, and application services handle business workflows.

This separation brings clear advantages:

Scalability comes standard when logic and data can scale independently.

Innovation moves faster because logic changes don't require database rewrites or coordination delays.

Operational overhead drops thanks to managed services that eliminate manual tuning and upkeep.

Reliability improves through clearer separation of components, making failures easier to isolate and recover from.

Future-proofing is built in since decoupled logic can evolve with your stack without being tied to a legacy database engine.

As a result, migrating stored procedures is now a design decision. Where the logic goes impacts both scalability and maintainability just as much as where the data goes.

When an organization chooses a Google Cloud target, it is implicitly answering questions like:

- Should business logic stay in the database or move out?
- Will logic run as analytics workflows, application services, or batch pipelines?
- How tightly should logic be coupled to the data platform?

Different Google Cloud databases answer those questions very differently.

This is why Google Cloud migrations start by aligning business logic with the target architecture, not the other way around.

When teams evaluate where their business logic truly belongs, they often arrive at very different Google Cloud platforms.

A team moving from Oracle to AlloyDB may decide to preserve database-centric logic while modernizing its structure and performance. Another team moving from SQL Server to BigQuery must accept that the database will no longer run operational logic at all. A third team adopting Cloud Spanner must redesign the system so business rules live entirely outside the database.

Each path can be successful but only if the implications are understood upfront.

In the next section, we'll introduce three migration case studies that illustrate how these choices play out in practice, and why different companies landed on different solutions.

The Three Migration Scenarios (At a Glance)

Industry	Legacy Platform	Key Business Challenge	Google Cloud Target	What Happens to Stored Procedures
Financial Software	Oracle	Preserve business-critical logic while reducing cost and modernizing performance	AlloyDB	Rewritten and modernized; logic remains in the database
Insurance	SQL Server	Modernize analytics platform and decouple logic from infrastructure	BigQuery	Rebuilt as analytics pipelines and workflows
Healthcare	DB2	Achieve global scalability and system resiliency without database-embedded logic	Cloud Spanner	Removed from the database and re-architected into application services

Let's look at the first of these examples: a regulated financial software provider modernizing from Oracle to AlloyDB.

Case Study 1

Financial Software

Modernizing from Oracle to AlloyDB to preserve critical logic and reduce operational costs.

This first example comes from the regulated financial software industry. The organization operated a mature software platform used for regulatory and financial reporting, built over many years on an on-premises Oracle database.

Hundreds of stored procedures handled batch processing, data validation, rule enforcement, and report preparation. These procedures were written in Oracle's PL/SQL and deeply intertwined with the data model and job scheduling mechanisms.

While the platform worked well, its high operating costs, limited scalability, and reliance on proprietary Oracle features had begun to hold the business back. These limitations were slowing client onboarding, restricting growth opportunities, and making it harder to ensure long-term continuity and resilience.

The Migration Goal

The organization's objective was modernization not a full architectural rewrite.

They wanted to:

Reduce licensing and operational costs

Move to a managed, cloud-native database

Preserve existing business behavior and processing logic

Avoid disrupting downstream applications and customers

This made AlloyDB a natural target. As a PostgreSQL-compatible managed database, AlloyDB supports stored procedures and transactional workloads while offering the operational benefits of Google Cloud.

The Stored Procedure Challenge

While AlloyDB supports stored procedures, it does not support Oracle's PL/SQL. This meant the logic could not be migrated directly.

Every procedure had to be:

- Analyzed to understand its true business purpose
- Rewritten in PostgreSQL-compatible procedural language
- Validated to ensure results matched the legacy system

In many cases, procedures had grown over time to serve multiple purposes, operational processing, reporting, and exception handling, all within a single block of code. Untangling this logic was one of the most time-consuming parts of the migration.

How the Logic Was Modernized

At the time of the migration, no commercial off-the-shelf (COTS) or AI-driven tools were available that could reliably translate complex Oracle logic to PostgreSQL in a way that preserved business behavior.

As a result, SoftServe developed custom migration tooling to support the process. The migration followed a controlled, phased approach:

Stored procedures were categorized by function (core processing, reporting support, maintenance jobs)

Critical business logic was rewritten and optimized for AlloyDB's execution model

Supporting jobs and batch processes were adjusted to fit cloud-native scheduling and deployment practices

Continuous testing ensured outputs matched Oracle results before cutover

Where appropriate, some logic was simplified or split into clearer components, improving maintainability without changing behavior.

Key Takeaways

By migrating to AlloyDB, the organization achieved a modernized database platform while preserving the core logic their business depended on. The transition reduced infrastructure complexity and licensing costs, while setting the foundation for future cloud-native improvements.

As a result, the business is now better positioned to scale operations, onboard new clients more efficiently, and evolve its platform without being constrained by legacy database limitations.

This approach works well when organizations want to modernize incrementally and retain a database-centric architecture.

In the next section, we'll examine a very different scenario: migrating from SQL Server to BigQuery, where stored procedures must be rebuilt outside the database entirely.

Case Study 2

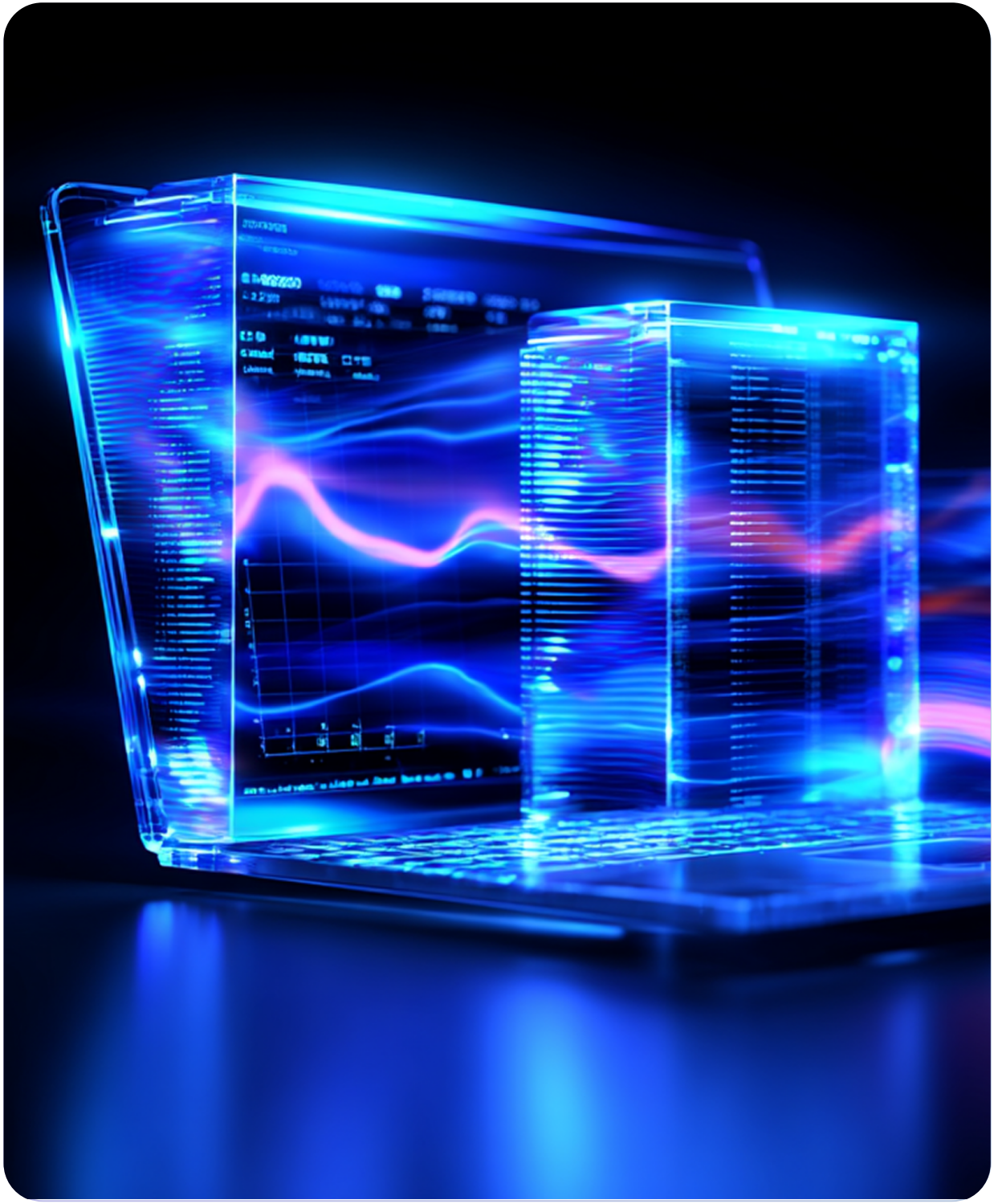
Insurance

Modernizing from SQL Server to BigQuery to rebuild analytics on a scalable, cloud-native foundation.

The second example comes from the insurance and reinsurance industry. The organization operated a mature reporting and analytics environment built on Microsoft SQL Server, with heavy use of T-SQL stored procedures.

Unlike the first case, these stored procedures were not primarily supporting transactional workloads. Instead, they formed the backbone of the reporting layer, transforming raw operational data, calculating metrics, and preparing datasets consumed by business intelligence tools.

Over time, the platform became harder to maintain. Multiple SQL Server instances struggled with growing data volumes, leading to reliability issues and rising operational overhead. Reporting couldn't scale easily, and workloads were pushing the system to its limits. The team decided it was time to move to Google Cloud.



The Migration Goal

The objective was not to recreate the existing system in the cloud, but to modernize analytics entirely.

Key goals included:

Reducing infrastructure and licensing

Improving data freshness and scalability

Moving toward a cloud-native analytics architecture

Preserving reporting results and business meaning

BigQuery was selected as the target platform because it is purpose-built for large-scale analytics and integrates naturally with Google Cloud's data ecosystem.

The Stored Procedure Challenge

BigQuery is purpose-built for large-scale analytics, designed to process massive datasets quickly using a serverless, distributed architecture. While it can support procedural business logic, that's not what it was designed for. As a result, the T-SQL stored procedures behind the legacy reporting layer couldn't be migrated directly. Instead, they were redesigned as part of a modern analytics pipeline built around BigQuery's strengths.

This required a mindset shift.

Instead of asking "How do we convert these procedures?", the team had to ask:

- What transformations are these procedures really performing?
- Which steps belong to analytics pipelines rather than the database?
- How should workflows be orchestrated in a cloud-native way?

How the Logic Was Rebuilt

The migration focused on extracting intent from the stored procedures and rebuilding that intent using analytics-native components:

Raw operational data was ingested into BigQuery on a near–real-time basis

Transformation logic previously embedded in T-SQL procedures was rewritten using BigQuery SQL

Dataform was used to manage dependencies, transformations, and versioning

Workflow orchestration replaced procedural execution order

Existing BI tools were repointed to the new BigQuery datasets

This approach decoupled analytics logic from any single database instance and introduced modern data engineering practices such as CI/CD, version control, and automated testing.

As a result, reporting logic became easier to maintain, faster to update, and less dependent on a specific platform, giving teams more flexibility to evolve their analytics workflows over time.

Key Takeaways

The result was a more scalable, cost-effective analytics platform that preserved reporting accuracy while significantly reducing operational overhead. Reporting logic became more transparent, easier to maintain, and better aligned with cloud-native data practices.

This case demonstrates a clean break from database-centric logic, a sharp contrast to the AlloyDB approach.

In the next section, we'll look at a healthcare organization migrating from DB2 to Cloud Spanner, where stored procedures are not supported at all and logic must move entirely into the application layer.

Case Study 3

Healthcare

Modernizing from DB2 to Cloud Spanner to enable global scale and rearchitect application logic.

The third example comes from the healthcare industry. The organization operated a centralized clinical platform supporting thousands of locations and tens of thousands of users, originally built on a DB2 database.

In this environment, stored procedures played a critical role. They enforced data integrity, coordinated workflows, supported operational reporting, and helped ensure consistent behavior across the system. The database was deeply embedded in the application's day-to-day operation.

As the platform grew to support thousands of clinics across the U.S., the limits of the existing architecture became clear. Managing and maintaining distributed DB2 instances created operational overhead and made it difficult to scale efficiently. Migrating to Google Cloud and adopting Cloud Spanner as the core database allowed the organization to consolidate infrastructure and achieve a single source of truth

The Migration Goal

The goal was not simply to move data to the cloud, but to enable a highly available, globally scalable platform capable of supporting thousands of clinics with consistent performance.

Key objectives included:

Horizontal scalability without operational bottlenecks

Strong consistency across distributed locations

Improved reliability and disaster recovery

A foundation for long-term platform evolution

Cloud Spanner was selected because it delivers these capabilities natively, but it comes with a significant architectural constraint.

The Stored Procedure Constraint

Cloud Spanner does not support stored procedures by design. It's built for global scale, and high availability, all while minimizing operational complexity. Embedding procedural logic in the database would work against these goals by tightly coupling business rules to the data layer.

Instead, Spanner encourages a clean separation of concerns. By keeping logic in the application layer, it simplifies cross-region scaling and keeps core database operations fast and reliable.

All logic that previously lived inside DB2 stored procedures had to be relocated. There was no option to translate or rewrite procedures within the database. Instead, the system needed to be redesigned so that business logic lived outside the database entirely.

How the Architecture Was Transformed

The migration required a shift to a service-oriented, cloud-native architecture:

Business rules and workflows were moved into application-level services

A microservices architecture was introduced to isolate domains and responsibilities

Event-driven patterns were used to coordinate processes across services

Data access logic was simplified to focus on persistence and consistency

New validation and auditing mechanisms ensured correctness without database procedures

Rather than relying on the database to orchestrate behavior, the application layer became the source of truth for business logic.

Key Takeaways

The result was a platform designed for scale, resilience, and long-term flexibility. While the upfront effort was higher than in the previous cases, the architecture aligned naturally with Google Cloud's strengths and positioned the organization for future growth.

This case represents the most dramatic shift in how stored procedures are handled but also the clearest example of how cloud-native design can reveal new capabilities.

The Leadership Takeaway

There's no single right way to handle stored procedures in Google Cloud.

Your Google Cloud destination largely determines what happens to your logic. Stored procedures can't simply be lifted and shifted. They may need to be rewritten, rebuilt, or moved out of the database entirely.

This is an opportunity to modernize. Removing stored procedures allows organizations to decouple business logic from legacy infrastructure, making systems easier to scale, easier to maintain, and faster to change. It opens the door to adopting CI/CD, modern observability, automated testing, and cloud-native deployment practices, all of which help businesses move faster, reduce risk, and future-proof their architecture.

Choosing a Google Cloud platform is a database decision and it's a decision about where your business logic will live going forward.

Teams that understand this early move faster and modernize with confidence. Those that don't often discover the implications too late, slowing down the migration, and making it more expensive.

The most successful migrations don't ask,

> ***How do we move our stored procedures?***

They ask,

> ***Where does this logic belong in the future?***

Why Experience Matters in Stored-Procedure-Heavy Migrations

Migrations that involve a lot of stored procedures are rarely straightforward. While stored procedures can provide clear value, such as keeping logic close to the data or improving performance in specific cases, they also tightly couple business rules to a specific platform. That's why getting them right during a migration requires understanding why the system was built that way in the first place.

That's the lens SoftServe brings to Google Cloud migrations.

Instead of starting with a target database and forcing everything into it, the focus starts with the logic itself. What does it do? What truly matters to the business? What needs to stay exactly the same, and what can be modernized? Answering those questions early changes how the entire migration unfolds.

As the examples in this paper show, there isn't a single "correct" approach. Sometimes logic is translated and kept in the database. Sometimes it's rebuilt as analytics workflows. Other times, it moves into application services. The difference between smooth migrations and costly ones is whether those choices are made intentionally.

In the end, experience matters not because of the tools used, but because it helps teams make the right decisions before problems appear.

Planning a Google Cloud migration with a lot of stored procedures? SoftServe can help you assess the logic, align it to the right architecture, and avoid costly redesigns later.

TALK TO SOFTSERVE

About SoftServe

SoftServe is a premier IT consulting and digital services provider. We expand the horizon of modern technologies to solve today's complex business challenges and achieve meaningful outcomes for our clients.

SoftServe & Google Cloud partnership

SoftServe, a Premier Google Cloud Service Partner, helps enterprises build the foundations for successful AI transformation and scale AI to meet bigger goals, faster. We work with organizations to break down silos, modernize applications, and turn fragmented data into a connected, AI-ready foundation. Whether you're scaling AI agents across your workforce or improving data quality for smarter, faster decisions, we bring the strategy, expertise, and technology to get you there.

Google Cloud
PREMIER
Services Partner

softserve

Contact

NORTH AMERICAN HQ

201 W 5th Street, Suite 1550
Austin, TX 78701
+1 866 687 3588 (USA)
+1 647 948 7638 (Canada)

EMEA HQ

30 Cannon Street
London EC4 6XH
United Kingdom
+44 333 006 4341

Social Links

info@softserveinc.com
www.softserveinc.com

softserve | Google Cloud