SIT378 GCP database

Kota Ishikawa

ID: 220530761

About GCP

Google Cloud Platform (GCP) was built to provide a set of computing resources. Google Cloud Platform (GCP) offers a wide range of computing resources, including database services. Database services are one of them. Efficient, flexible, and capable of processing modern data with superior performance, GCP is a platform solution for hosting geographically dispersed data. GCP offers three reference architectures for global data distribution: hybrid, multi-cloud, and geo-distributed. These architectures should be considered when choosing a Google database service. There are also many things to consider when choosing a Google database service, such as data type and size, latency, throughput, scalability, and IOPs.

GCP primarily offers three different reference architecture models for global data distribution.

1) Single

— The simplest of all deployment models, you can either create a new cloud database at Google or deploy the database by “lifting and shifting” existing workloads.

2) Hybrid — These types of deployments are useful when you have applications in the cloud that need to access on-premises databases, or vice versa.

There are three main factors to consider when deploying a hybrid model (some data is on Google Cloud and some data is on-premises).

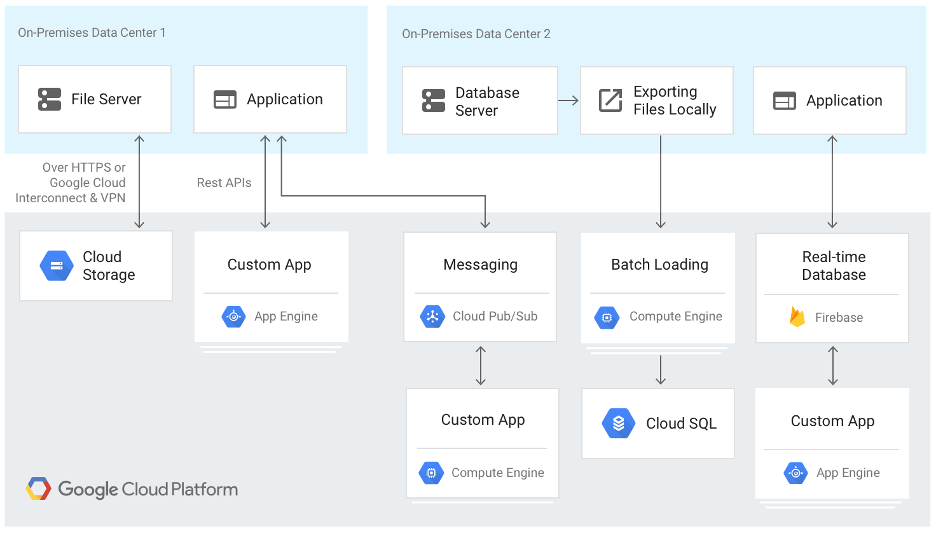
Master database: First and foremost, you need to decide whether to store your master database on-premises or in the cloud. If you choose cloud, your GCP resources act as a data hub for your on-premises resources, while if you choose on-premises, your on-premises resources sync data to the cloud for remote use or backup.

Managed Services: Available for resources in the cloud, these services consist of scalability, redundancy, and automatic backups. However, you have the option of using a third-party managed service.

Portability: Data portability is also affected based on the type of data store you choose. Cross-platform stores such as MySQL should be considered to ensure reliable and consistent data transfer.

3) Multi-cloud — These types of deployments help distribute databases effectively and create multiple fail-safes. This allows you to combine databases deployed on Google Cloud with database services from other cloud providers, giving you a unique set of services. cloud functionality.

　The following diagram shows a high-performance system using Google Cloud and an on-premises system.



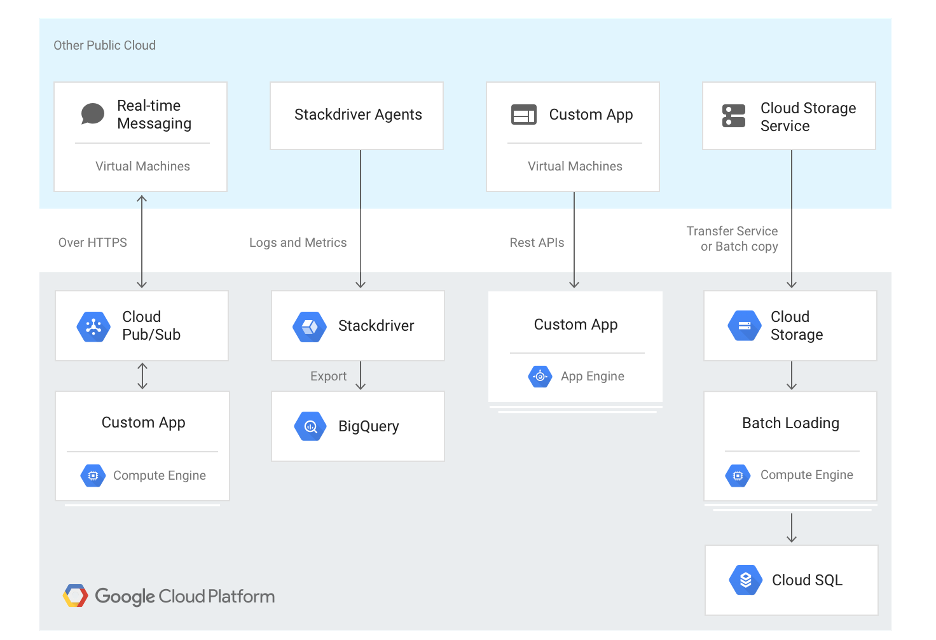
<https://bluexp.netapp.com/blog/gcp-cvo-blg-google-cloud-database-the-right-service-for-your-workloads#:~:text=Google%20Cloud%20Platform%20>

There are two main factors to consider when deploying this model.

Integration: Allows client systems to seamlessly access databases regardless of the cloud in which they are deployed. For example, use open-source client libraries to make databases seamlessly available across clouds.

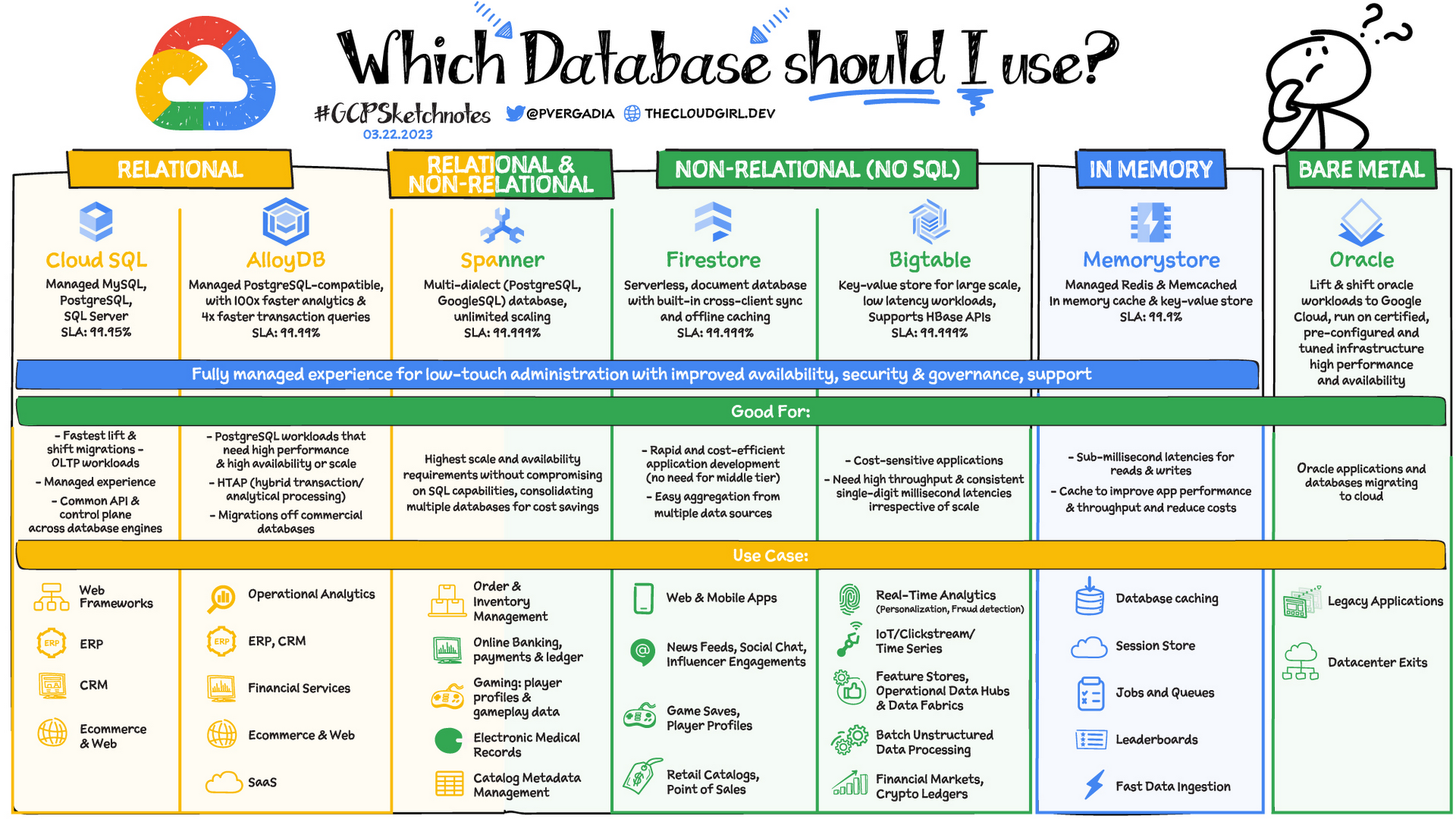
Migration: With multiple cloud providers, you may need to migrate data between clouds using database replication tools or export/import processes. The Google Storage Transfer service is one tool that can help you migrate your database.

The following diagram shows a multi-cloud deployment that includes GCP and another public cloud provider.



<https://bluexp.netapp.com/blog/gcp-cvo-blg-google-cloud-database-the-right-service-for-your-workloads#:~:text=Google%20Cloud%20Platform%20>

Explanation of a database that can be used with a simple GCP



<https://cloud.google.com/blog/topics/developers-practitioners/your-google-cloud-database-options-explained?hl=en>

Google cloud database service

GCP offers multiple Google Cloud database services to choose from. Below is an introduction to each.

cloud SQL

Cloud SQL is a fully managed relational Google Cloud database service compatible with SQL Server, MySQL and PostgreSQL. It includes automated backup, data replication, and disaster recovery capabilities to ensure high availability and resilience. This service can be integrated with Compute Engine, App Engine, BigQuery, and Kubernetes.

Common use cases for Cloud SQL are:

• Lift and shift on-premises SQL databases to the cloud

• Large-scale SQL data analysis

• Support for content management system (CMS) data storage and scalability

• Managing databases using Infrastructure as Code (IaC)

• Develop and deploy containerized applications and microservices

cloud spanner

Cloud Spanner is another fully managed relational Google Cloud database service. Unlike Cloud SQL, it focuses on combining the advantages of relational structure and non-relational scalability. It provides strong line-to-line consistency and high-performance operation. Features include automatic replication, built-in security, and multilingual support.

Use cases for Cloud Spanner include:

• Supply chain management and manufacturing

• financial trading, analysis and forecasting;

• Logistics and transportation

BigQuery

BigQuery is a fully managed serverless data warehouse. You can use it to perform data analysis and query streaming data via SQL. This service includes a built-in Data Transfer Service to help migrate data from on-premises resources such as Teradata.

BigQuery includes capabilities for machine learning, business intelligence, and geospatial analytics. These capabilities are provided through BigQuery ML, BI Engine, and GIS.

BigQuery use cases include:

• Process analysis and optimization

• Big data processing and analysis

• Machine learning based behavioral analysis and prediction

• Data warehouse modernization

cloud big table

Cloud Bigtable is a fully managed NoSQL Google Cloud database service. Designed for large-scale operational and analytical workloads. Cloud Bigtable includes features for high availability, zero downtime configuration changes, and sub-10ms latency. Integrate with a variety of tools including Hadoop, Apache tools like TensorFlow, and Google Cloud services like BigQuery.

Use cases for Cloud Bigtable include:

• Financial analysis and forecasting

• Internet of Things (IoT) data ingestion, processing, and analysis

• Marketing applications including hyper-personalization

cloud firestore

Cloud Firestore is a fully managed serverless NoSQL Google Cloud database designed for developing serverless apps. You can use it to store, sync, and query data for web, mobile, and IoT applications. This includes features for offline support, live sync, and built-in security. You can integrate Firestore with GCP's mobile development platform, Firebase, to make it easier to create and manage apps.

Use cases for Cloud Firestore include:

• Mobile and web applications with both online and offline functionality

• Multi-user collaboration application

• Real-time analytics

• Social media applications

• Gaming forums and leaderboards

Firebase Realtime Database

Realtime Database is a NoSQL Google Cloud database that is part of the Firebase platform. It can store and sync data in real time and includes caching for offline use. Realtime Database can also implement declarative authentication that matches users by ID or pattern matching. It includes mobile and web software development kits (SDKs) to make app development easier and faster.

Use cases for Firebase Realtime Database include:

• Developing apps that work across devices

• Advertising optimization and personalization

• Third party payment processing

• Machine learning integration

cloud memory store

Cloud Memorystore is a fully managed in-memory Google Cloud data store. Designed to be secure, highly available, and scalable. Cloud Memorystore allows you to create application caches with sub-millisecond data access latency. Compatible with Memcached and Redis protocols.

Use cases for Cloud Memorystore include:

• Application lift-and-shift migration

• Machine learning applications

• Real-time analytics

• Low latency data caching and retrieval

How to choose a Google Cloud database service

Even after considering database options on Google Cloud, it can be difficult to decide which option is right for you. As you consider your options, keep in mind that many organizations need and benefit from multiple services. This allows you to optimize your implementation according to the capabilities of your database rather than trying to adapt your database service to suit all your needs.

Cloud SQL

Cloud SQL is a good option if you want relational database functionality, but don't need more than 10 TB of storage capacity or 4000 concurrent connections. You also need on-premise management skills.

Cloud Spanner

Cloud Spanner is a good option if you plan to use large amounts of data (over 10 TB) and need transactional consistency. Also suitable if you want to use sharding for higher throughput and accessibility.

Cloud Scanner is a better choice than Cloud SQL if you know or think you will eventually need to be able to scale your Google Cloud databases horizontally. If you started with Cloud SQL and eventually need to migrate to Cloud Spanner, be prepared to rewrite your application in addition to migrating your database.

Cloud Firestore/Datastore

Cloud Firestore or Datastore are good options if you plan to focus on app development and need live sync and offline support.

If you need to store unstructured data in JSON documents, Cloud Datastore is the preferred option. This is in comparison to when you need to store structured data. Cloud Spanner is recommended in this case.

An additional factor to consider is whether you need atomicity, consistency, isolation, and durability (ACID) compliance. In that case, Cloud Spanner should be your choice, as Cloud Datastore only offers atomic and durable transactions.

Cloud Bigtable

Cloud Bigtable is a good option if you have large amounts of single-key data. Especially suitable for low-latency, high-throughput workloads.

Cloud Bigtable is preferred over Cloud Spanner when you need to perform single-region analysis. However, if you need multi-region operations, Cloud Spanner is the preferred solution. For example, Cloud Bigtable is a good option for time series apps built for DevOps monitoring. Cloud Spanner, on the other hand, is the preferred option for infrastructure monitoring platforms designed for Software as a Service (SaaS).

Cloud Memorystore

Cloud Memorystore is a good option if you are working with key-value datasets and transaction latency is your primary concern.

Cloud Memorystore should be your choice if you don't need disk-based data persistence and are using the service for caching only. However, if you are concerned about issues such as cache-to-database consistency or stream processing, Cloud Bigtable should be your choice. Similarly, if the amount of data is too large to fit in memory, Cloud Memorystore is not the best choice.

When divided into relational and non-relational

relational database

A relational database stores information in tables, rows, and columns. This is usually best for structured data. As a result, it is used for applications where the structure of the data does not change frequently. SQL (Structured Query Language) is used when interacting with most relational databases. Provides ACID consistency mode for data. This means:

• Atomic: All operations within the transaction succeed or the operations are rolled back.

• Consistency: The database is structurally sound when transactions are completed.

• Isolation: Transactions do not conflict with each other. Competing access to data is coordinated by the database so that transactions appear to run sequentially.

• Durable: The result of applying a transaction is durable even in the event of failure.

These characteristics make relational databases used for applications that require high precision and for transactional queries such as financial and retail transactions. Example: In banking, when a customer makes a request to transfer funds, we need to ensure that the transaction is possible and indeed done with the latest account balance. Errors or resubmitting requests are probably fine in this case.

Google Cloud offers three relational database options: Cloud SQL, Cloud Spanner, and Bare Metal Solution.

• Cloud SQL : Provides MySQL, PostgreSQL, and SQL Server databases managed by Google Cloud. Reduce maintenance costs and automate database provisioning, storage capacity management, backups, out-of-the-box high availability and disaster recovery/failover. For these reasons, it is ideal for general-purpose web frameworks, CRM, ERP, SaaS, and e-commerce applications.

• AlloyDB: His fully managed PostgreSQL-compatible database service for the most demanding enterprise database workloads. It combines the best of Google with his one of the most popular open source database engines, PostgreSQL, for great performance, scale and availability. AlloyDB is 4x faster than standard PostgreSQL for transactional workloads and up to 100x faster for analytical queries. A machine learning-enabled autopilot system simplifies database management. AlloyDB offers transparent and predictable pricing without expensive licensing or opaque I/O fees.

• Cloud Spanner: Cloud Spanner is an enterprise-grade, globally distributed, strongly consistent database that offers up to 99.999% availability, purpose-built to combine the benefits of relational database structures with non-relational horizontal scale. increase. It is a unique database that combines ACID transactions, SQL queries, and relational constructs with the scalability typically associated with non-relational or NoSQL databases. As a result, Spanner is ideal for applications that require unlimited scaling capabilities with strong consistency and high availability, such as gaming, payment solutions, global financial ledgers, retail banking, and inventory management.

• Bare Metal Solution: Provides hardware to run specialized workloads with low latency on Google Cloud. This is especially useful if you have an Oracle database that you want to lift and shift to Google Cloud. This enables data center retirement and paves the way for modernizing legacy applications.

Non-relational database

Non-relational databases (or NoSQL databases) store complex, unstructured data in non-tabular formats, such as documents. Non-relational databases are often used when large amounts of complex and diverse data need to be organized, or when the structure of the data is constantly evolving to meet new business requirements. Queries run faster because they don't have to access multiple tables to get answers, unlike relational databases. It is ideal for storing data that may change frequently or for applications that process different types of data. For example, a clothing store might have its own document with all the information such as size, brand, color, etc., and a database to which parameters such as sleeve size and collar could be added later.

Characteristics that make NoSQL databases fast:

• They are typically optimized for specific workload patterns (ie key values, graphs, wide columns).

• Horizontal scaling, usually using range or hash distributions

• Eventual Consistency: Many NoSQL stores typically exhibit consistency at a later point in time (for example, read delays). However, Firestore uniquely provides strong global consistency.

• Transactions: Most NoSQL stores do not support cross-shard transactions or flexible isolation modes. However, Firestore uniquely serves his ACID transactions across shards using serializable isolation.

These characteristics make nonrelational databases used in applications that require large scale, reliability, availability, and frequent data changes. Some relational databases scale horizontally easily by adding servers, as opposed to growing vertically by increasing the size of the machine. as the data grows. However, some relational databases, such as Cloud Spanner, support scale-out and strong consistency.

Non-relational databases can store a wide variety of unstructured data, such as documents, key values, graphs, and wide columns. Google Cloud's non-relational database options are:

• Document database: stores information as documents (in formats such as JSON and XML). example:

• Key-value store: Uses records identified by unique keys to group related data in collections for easy retrieval. A key-value store is structured enough to reflect the values of a relational database while retaining the benefits of NoSQL. Examples: Bigtable, Memorystore

• In-memory database: A purpose-built database that relies primarily on memory for data storage. They are designed to provide minimal response time by eliminating the need to access disks. They are ideal for applications that require microsecond response times and can experience large spikes in traffic. Example: memory store

• Wide databases: Use a tabular format, but the names and formats of data can vary widely in each row, even in the same table. It has some basic structure while maintaining a lot of flexibility. example:

• Graph databases: Use graph structures to define relationships between stored data points. Helps identify patterns in unstructured and semi-structured information. Example: JanusGraph

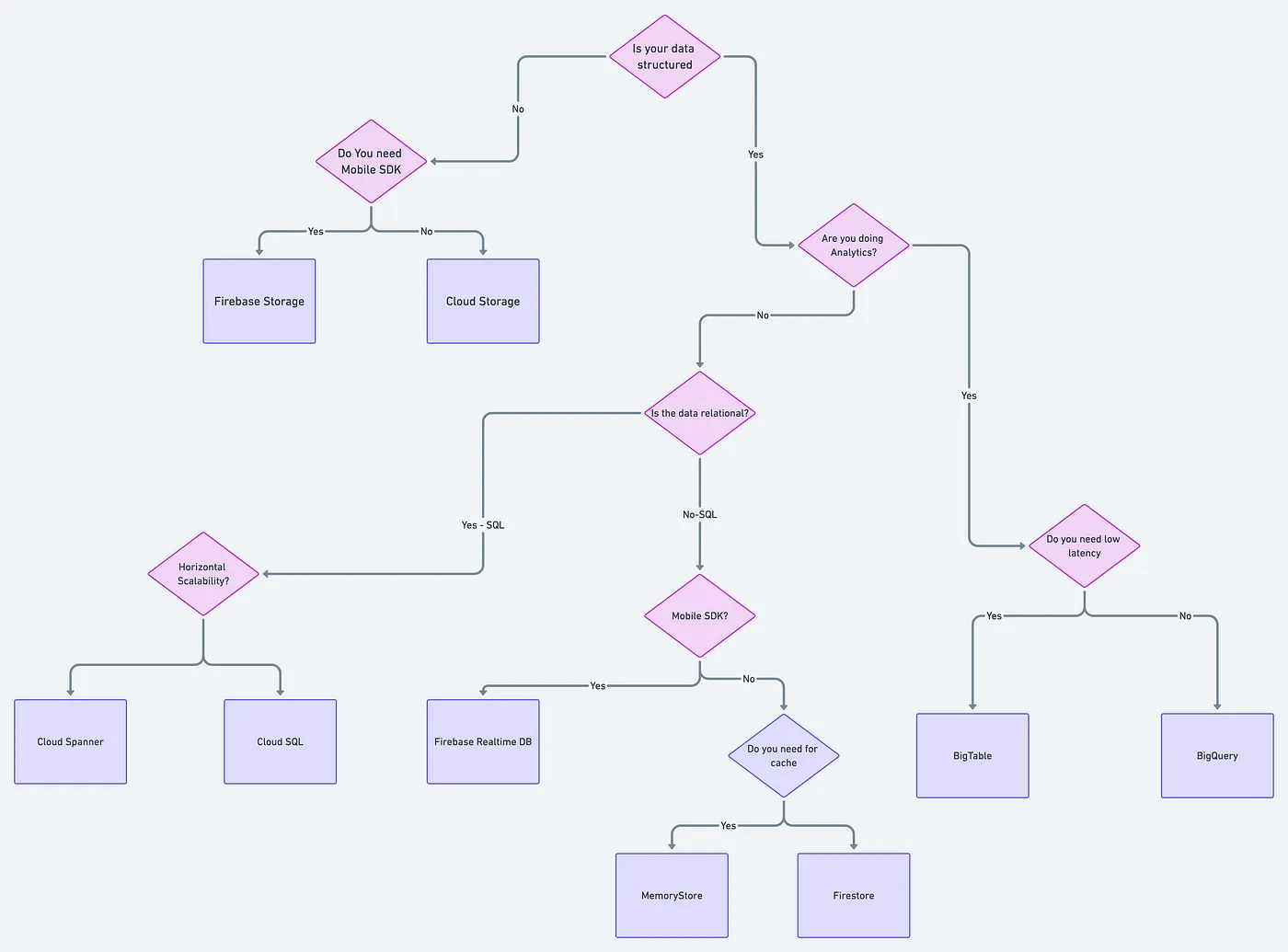
Google Cloud has three non-relational databases.

• Firestore: A serverless document database that scales on demand, has strong consistency, provides up to 99.999% availability, and serves as a backend as a service. DBaaS optimized for building applications. Perfect for all general-purpose use cases such as e-commerce, gaming, IoT, real-time dashboards, and more. Firestore allows users to interact and collaborate on live and offline data, making it ideal for real-time applications and mobile apps.

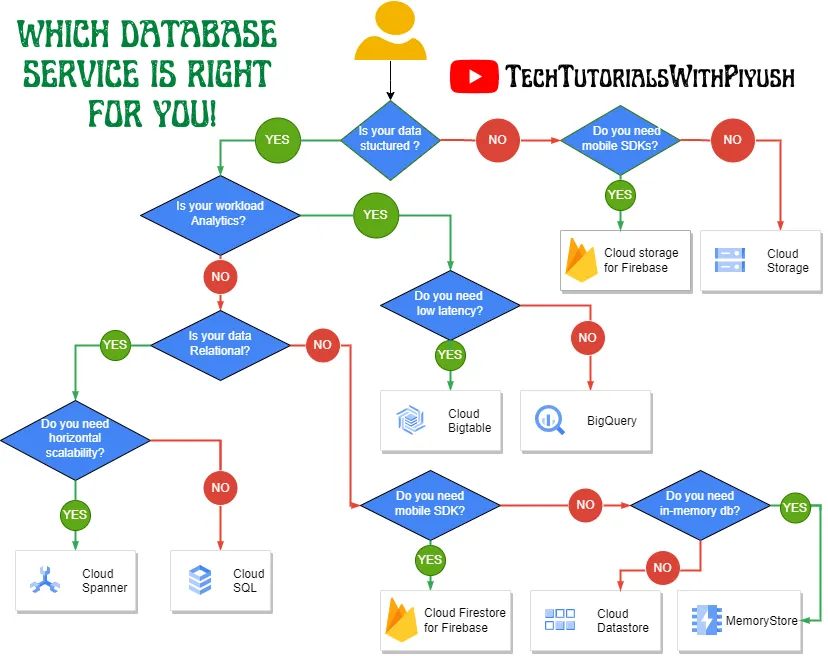
• Cloud Bigtable: Cloud Bigtable is a sparsely populated table that scales to billions of rows and thousands of columns, and can store terabytes or petabytes of data. This is ideal for storing very large amounts of single-key data with very low latency. It supports high read and write throughput with sub-millisecond latency, making it an ideal data source for MapReduce operations. It also supports the open source HBase API standard for easy integration with the Apache and Google Cloud ecosystems including HBase, Beam, Hadoop and Spark.

• Memorystore: Memorystore is a fully managed in-memory datastore service for Redis and Memcached on Google Cloud. It is ideal for in-memory and transient data stores and automates the complex tasks of provisioning, replication, failover, and patching so you can spend more time coding. Offering extremely low latency and high performance, Memorystore is ideal for web and mobile, gaming, leaderboard, social, chat and news feed applications.

chart of which to choose

****

[**https://towardsdatascience.com/how-to-choose-the-right-google-cloud-platform-database-a223f4d7482f**](https://towardsdatascience.com/how-to-choose-the-right-google-cloud-platform-database-a223f4d7482f)

****

[**https://medium.com/google-cloud/choose-the-right-database-service-in-gcp-8e3803245e1d**](https://medium.com/google-cloud/choose-the-right-database-service-in-gcp-8e3803245e1d)

**Reference list**

**Anuji Syal(2021) HOW to choose the right google Cloud Platform Database**

[**https://bluexp.netapp.com/blog/gcp-cvo-blg-google-cloud-database-the-right-service-for-your-workloads#:~:text=Google%20Cloud%20Platform%20**](https://bluexp.netapp.com/blog/gcp-cvo-blg-google-cloud-database-the-right-service-for-your-workloads#:~:text=Google%20Cloud%20Platform%20)

**Piyush Sachdeva(2022) Choose The Right Database Services in GCP**

[**https://medium.com/google-cloud/choose-the-right-database-service-in-gcp-8e3803245e1d**](https://medium.com/google-cloud/choose-the-right-database-service-in-gcp-8e3803245e1d)

**(2021) Google Cloud Database Services Explained: How to Choose the Right service for your workloads**

[**https://bluexp.netapp.com/blog/gcp-cvo-blg-google-cloud-database-the-right-service-for-your-workloads#:~:text=Google%20Cloud%20Platform%20**](https://bluexp.netapp.com/blog/gcp-cvo-blg-google-cloud-database-the-right-service-for-your-workloads#:~:text=Google%20Cloud%20Platform%20)

**(2021) Your Google Cloud database option, explained**

[**https://cloud.google.com/blog/topics/developers-practitioners/your-google-cloud-database-options-explained?hl=en**](https://cloud.google.com/blog/topics/developers-practitioners/your-google-cloud-database-options-explained?hl=en)