*Short communication*

**Comparative Insights on Centralized and Individual Models Using Snowflake and Google Big Query**

ABSTRACT

|  |
| --- |
| Data sharing plays a vital role in today’s digital ecosystem, allowing businesses, governments, and individuals to exchange information on a massive scale. Cloud-native data platforms have significantly transformed traditional data management practices by introducing scale architectures, decoupled storage and compute models, cost-efficiency and resource governance. This paper investigates two distinct paradigms, (Centralized and Individual) data sharing models. Centralized architecture offers consolidated governance, multi-tenant scalability, and advanced analytics enablement, whereas individual data control underlines autonomy, privacy-preserving access, and federated governance. The comparative explores both paradigms through the capabilities of two leading platforms: Snowflake and Google Big Query (GBQ). This analysis highlights operations implications, governance complexity and data collaboration potentials. This paper also covers the pathways for organizations to adopt hybrid data sharing strategies with balance agility, regulatory compliance and efficiency in multi cloud environment |

*Keywords: Data Sharing, Centralized Platforms, Data Sovereignty, Security, Data Governance, Information Management, Data Protection, Snowflake, GBQ*

1. INTRODUCTION

Data has become central to business operation and to making strategic decisions. Two common approaches have emerged: centralized platforms, where data is managed from a unified environment, and individual data control, where teams or departments govern their own data independently. Both models offer distinct benefits and pose unique challenges, particularly in terms of governance, efficiency, security, and flexibility. Cloud data platforms like Snowflake and GBQ provide robust infrastructure to support both approaches. Understanding how each platform enables centralized and decentralized data management is essential for businesses aiming to scale data operations while maintaining control and compliance.

**2. Methodology**

**2.1 Evaluation Criteria**

The comparative analysis is structured around several key dimensions of data sharing within cloud environments, focusing on the concept of centralized and individual data sharing approaches in Snowflake and Google Big Query (GBQ). It defines the evaluation criteria based on the best industry practices, platform documentation, explore tools, and insights gained through a comparative evaluation of both models from a technical and operational perspective.

**2.2 Data collection method**

To understand the real-world effectiveness, the data was collected from the organization using either Snowflake or Google Big Query, official product documentation in Snowflake Docs, Big Query Docs, Google Cloud IAM documentation and white papers, platform tutorials and professional certification training materials. These sources provided insight into supported features, architectures, and practical implementation patterns.

**3. Overview of Two Approaches**

**3.1 Centralized Data Sharing**

A Centralized data repository is shared with multiple consumers or tenants from a single location within platform (Snowflake account or Big Query Project or dataset). Instead of copying data, access is granted directly to the original source. This approach is a single source of truth, simplifies processes, reduces storage needs, and ensures consistent version of data for all consumers.

* Unified policy enforcement
* Streamlined governance
* Reduced duplication of data

**3.2 Individual (Controlled) Data Sharing**

This model allows each tenant an independent and protected environment at the cost of potential duplication and complexity. In Snowflake, this means by separate tables/schemas or accounts. In Big Query by separate datasets or projects. Data is not mingled; sharing is accomplished by individual datasets or zero-copy links for each party. This approach offers greater flexibility and customization but can result in duplicated data and inconsistent practices.

* Fine-grained IAM controls
* Autonomous sharing decisions
* Permission project or dataset

**3.3 Purpose of comparative analysis**

This comparison highlights how Snowflake and Google Big Query (GBQ) support these centralized and individual data sharing approaches. It helps organizations to assess the trade-offs, evaluate real-world use cases, and design a scalable, governed, cost-efficient, and flexible data architecture.

**3.4 Key Features of these Models**

* Data Control
* Data Governance & Security
* Compliance
* Data Sharing Complexity
* Customization & Cost Efficiency

**4. Snowflake Centralized Sharing Approach**

Snowflake is a modern cloud-based data platform that excels in scalable analytics and adapts centralized data sharing. Unlike traditional data warehouses, it engages a multi-cluster, data architecture with centralized storage, separating computing from storage to enable high performance, concurrency, and flexibility. One of Snowflake’s key features is its data sharing competence, allowing organizations to share live, queryable data in real-time without the need to copy or move data.

**4.1 Secure Data Sharing**

Secure Data Sharing enables providers to grant access to their data without making copies. Consumers can query the shared data directly and in real-time through their own Snowflake accounts.

* **Real-time updates:** Eliminates the need for ETL pipelines or file transfers
* **Read-only access**: Ensures consumers cannot modify the original data
* **No storage overhead**: Data remains in the provider’s account, avoiding additional storage costs
* **Governed sharing**: Robust controls like role-based access (RBAC), data masking, and tagging are applied to maintain security and compliance

**Use Case Example**: A retail company shares real-time transaction summaries with a logistics partner to monitor inventory levels and optimize supply chain operations, ensuring instant and secure access to essential data.

**4.2 Share Object**

A Share functions as a metadata object in modern data-sharing ecosystems, acting as a gateway for controlled access to resources. It is designed to include:

* Key elements such as databases, schemas, tables, views, and user-defined functions (UDFs)
* Permissions are defined through access control grants, ensuring precise control over data usage

**4.3 Governance & Permissions**

**4.3.1 Snowflake Consumer Account**

In a centralized snowflake model, also referred to multi-tenant table, all tenant data resides within one account. Therefore, robust row-level or object-level security is essential to prevent data leaks. **Role-Based Access Control (RBAC)** and secure views to restrict data per tenant. Tenants query the data using standard SQL with no data ingestion required.

*4.3.1.1 Reader Account*

For external partners without a Snowflake account, the provider can create a managed Reader Account, giving them UI and SQL access with limited administrative overhead.

*4.3.1.2 Snowflake Marketplace*

The Snowflake Marketplace lets organizations publish and monetize data products or find external datasets. It offers open and commercial data sets, supports data-as-a-service models, and includes enriched data from finance, healthcare, and retail industries.

**4.4 Performance**

In a centralized model, snowflake uses separate compute clusters to handle each consumer queries, eliminating resource contention on the central data. Supports multi-cluster warehouses with additional auto-scale compute clusters to handle high concurrency, ensuring scalable performance Providers must size warehouses for peak loads or enable auto-scaling. If Consumers with their own Snowflake accounts use their own warehouses, isolating compute load while maintaining efficient data access via a unified storage layer.

**4.5 Cost**

Centralizing data in Snowflake is storage efficient data is stored once. Moreover, with Secure Data Sharing, the consumer only query compute costs and does not incur storage costs. Providers can also execute queries on a central account, bearing compute costs but benefiting from shared resources. Overall, central sharing reduces data duplication costs and is more cost-effective when serving multiple consumers​.

**5. Google Big Query Centralized Sharing Approach**

Google Big Query (GBQ) is a cloud-based fully managed, serverless data warehouse that is capable of centralized data sharing. It allows organizations to analyze large datasets. With Big Query, teams, projects, and organizations can share data securely and efficiently without having to copy or move it around. In 2021, Google introduced **Analytics Hub**, a data exchange that functions similarly to Snowflake’s marketplace for sharing data on a scale.

**5.1 Dataset Sharing**

Datasets serve as a core fundamental unit for data sharing in Big Query. Using **IAM roles**, access to datasets and their contents (tables, views, models) can be granted to specific users, groups, or service accounts.

* **Viewer** (roles/bigquery.dataViewer) – Grants Read-only access
* **Editor** (roles/bigquery.dataEditor) – Allows both Read/write access
* **Owner** (roles/bigquery.dataOwner) – Full control including administrative access

IAM can be applied at various levels:

* Project level
* Dataset level
* Table level

**5.2 Governance & Permissions**

**5.2.1 Authorized Views**

Authorized Views in Big Query enable a secure and efficient dataaccess management by exposing only specific fields or rows to consumers.

* Isolated Query Interface: Consumers query the view rather than the underlying table and table never directly exposed.
* Suitable for implementing row-level or column-level security.

**Use Case**: A retailer shares aggregated sales data with partners, masking sensitive fields to enable secure analytics on trends and inventory.

**5.2.2 Row-Level & Column-Level Security**

Big Query offers fine-grained access control via:

* Row-Level Security (RLS): Filters data based on user attributes.
* Column-Level Security (CLS): Restricts access to specific columns using policy tags managed in Data Catalog.

These features enhance compliance and governance across multi-tenant or regulated environments.

**5.2.3 Cross-Project Data Sharing**

Big Query provides cross-project controls like VPC Service Controls to prevent unauthorized data access, which is useful in multi-tenant setups. In a centralized model, the provider retains full control of data, and consumers have read-only access​. This central governance ensures data consistency and security.

**5.3 Performance**

Big Query’s multi-tenant design supports concurrent users by auto-allocating slots and scales transparently for read-only workloads. Resource quotas and slot contention may arise with heavy queries, managed via Reservations to allocate dedicated slots per consumer or workload. Fair scheduling and a robust backend handle open data sharing, with scaling or regional data splits addressing extreme workloads.

**5.4 Cost**

In Big Query, providers pay for storage, while consumers bear query costs. On-demand pricing isolates query expenses for each project. Providers can monetize via Cloud Marketplace subscriptions, shifting storage costs to consumers. Clustering or partitioning optimizes query efficiency. Unlike Snowflake, Big Query lacks predefined warehouses, so expensive queries impact the billing project directly, encouraging providers to enforce consumer-side querying via linked datasets.

**6. Snowflake Individual Sharing Approach**

The individual model isolating each tenant a separate own data environment, with dedicated pipelines or automated delivery. Snowflakes support this approach for better security and customization.

**6.1 Snowflake’s Individual Tenant Isolation Approach**

Twp main patterns for tenant isolation **Object Per Tenant (OPT)** and **Account Per Tenant (APT). OPT** – Single account Isolation. Data in separate tables​, schema or database within a single snowflake account. **APT** –Full Isolation. Fully in separate accounts with isolated storage and compute.

**6.2 Governance & Security**

Snowflake’s unique model ensures robust isolation with separate schemas or accounts. In an APT design, each customer gets a fully isolated environment ideal for strict security requirements. Due to each account having its own RBAC namespace, the provider doesn’t need complex row-level security logic. Moreover, Snowflake’s Organizations feature facilities they manage multiple accounts centrally.

**6.3 Performance**

In Snowflake’s isolated model, each tenant can get their own compute resources, ensuring that other tenants do not affect performance. In APT setup fully separate workloads, while shared accounts can keep performance separate by using different warehouses. This method ensures consistent performance but may reduce efficiency if many small warehouses are underused.

**6.4 Cost**

Per-tenant isolation tracks cost and assignment, though overall expenses increase due to duplicated storage and idle compute. An APT works best for fewer tenants but may potentially increase administrative overhead. However, bulk deals and easy tenant offboarding help balance the trade-offs.

**7. Big Query Individual Sharing Approach**

Big Query using an isolated model typically get **dataset-per-tenant** within a single project or across a few projects. Google Cloud’s SaaS best practice is to “configure datasets for each tenant” to handle scaling to thousands of tenants ensuring better data isolation, access control, and performance.

**7.1 Dataset per Tenant**

Creating a single GCP project for all customers’ datasets, each dataset tagged by specific tenants. The ingest pipeline writes data into respective tables for each customer. This model isolates storage, schema, access controls, and usage tracking per tenant. Based on IAM role the access can be granted to a service account or group for each tenant.

**7.2 Governance & Security**

Tenants are strongly isolated by design, preventing access to each other’s data. IAM policies are strictly enforced at the dataset level, serving as a hard boundary in GCP. For enhanced compliance, optional features like VPC Service Controls or Customer-Managed Encryption Keys (CMEK) can be implemented.

**7.3 Performance**

Big Query handles isolated datasets efficiently. Data is partitioned by dataset queries from one tenant will only scan the tenant's tables. Specific queries reduce the need for complex SQL filters. Heavy users can be moved to dedicated projects with reserved slots. Google's fair scheduling prevents resource starvation, but heavy usage may consume shared slots. Providers can use reservations to divide the slot capacity between tenants or groups for stable performance.

**7.4. Cost**

With isolated datasets, storage increases per tenant, and costs are billed separately. If each tenant has X GB, 100 tenants mean 100\* X GB total, efficiently prevent cross-tenant cost spikes. Centralized storage offers better compression, the benefits in Big Query are minor. Google recommendation of one dataset per tenant provides a balance approach for a good SaaS.

**8. Comparative Platform Analysis**

**Table 1. Analysis the approaches on two platforms**

|  |  |  |
| --- | --- | --- |
| **Category** | **Centralized Sharing** | **Individual Sharing** |
| **Snowflake**  | **Big Query** | **Snowflake** | **Big Query** |
| **Data Residency** | Account | Dataset | Per tenant/ schema | Per tenant/dataset |
| **Authorization Mechanism** | Row-level access policies | IAM + column-level security | RBAC per tenant | IAM per dataset  |
| **Processing Isolation** | Shared computing (VW group) | Shared slot pools (fair scheduling) | Dedicated VW per tenant | Slot reservations per tenant |
| **Storage Utilization** | Highly efficient | Efficient  | Duplicate storage across schemas | May involve data redundancy |
| **Cost Allocation** | Storage paid by provider; Computing by Consumer | Storage paid by provider; Slots by Consumer | Cost per tenant (clear attribution, higher overhead) | Clear cost per tenant (project-level billing, labels) |
| **Scalability Threshold** | Supports millions of tenants  | Limited by 2500 views per dataset | Limited to 100 tenants; complex beyond that | Scales to thousands of tenants (datasets) |
| **Operational Automation** | Low | Low  | Medium–high  | Medium  |
| **Data Segregation** | Logical (row-level policies) | Logical (authorized views, column restrictions) | Strong (schema/account level) | Strong (dataset/project boundary) |
| **Compute Optimization** | Multi-cluster warehouses auto-scale for concurrency | Slot autoscaling under reservation model | Per-tenant compute warehouses | Reservations allow per-tenant slot pools |
| **Applicability** | SaaS platform uniform, data marketplace, shared apps | Public/shared data access, centralized analytics  | Compliance driven SaaS, tenant specific schema | High secured multi-tenant SaaS platform, isolation processing, compute isolation |
| **Compliance** | Requires precise row-level policy setup | Involved authorized view and permission setup | Easier to isolate per tenant, increases object-level administration | Clear isolation, simplified IAM control |

**9. Analysis and Discussion**

**9.1 Governance and Compliance**

**Snowflake** centralized model excels in regulated industries, and data access is strictly controlled and auditable. Features like object tagging, centralized access policies, and marketplace reduce complexity and streamline permission management.

Mainly, **Big Query** focuses on developer and sovereignty, suitable for data science or organizations prioritizing project-based access. Nevertheless, this approach increases the burden of maintaining consistency and auditability across departments.

**9.2 Performance and Operational Efficiency**

**Snowflake** architecture permits live sharing without copying data, reducing storage redundancy and data movement overhead. The platform is a multi-cluster shared data architecture that allows multiple users to access the data concurrently without any performance impact. Also auto scaling and optimization to manager workload.

**Big Query** may involve data replication or complex setup to share across projects, impacting query latency and cost efficiency due to increased latency and processing expenses. Moreover, depending on the dataset size and complexity, query performance might suffer due to data duplication.

**9.3 Ecosystem and Collaboration**

**Snowflake** has invested significantly in its **data collaboration ecosystem** including the Data Cloud and Marketplace.

**Big Query** is still evolving in this area, with its **Analytics Hub** being relatively new and less integrated.

****

**fig .1 Data Sharing Platform**

**10. Conclusion**

This study demonstrates that Snowflake’s centralized data sharing model offers a powerful and secure approach to enterprise-wide data collaboration, particularly suited for organizations with complex governance requirements.

In contrast, Google Big Query’s decentralized model favors flexibility and autonomy, ideal for teams operating in fast-paced environments or those with lighter governance needs.

**10.1 Recommendations**

* Choose Snowflake if you need:
	+ Unified and consistent data governance
	+ Multi-tenant data access with strong control
	+ Real-time data sharing without latency
* Choose Big Query if you prefer:
	+ Project level-based autonomy
	+ Seamless integration with Google Cloud-native tools

**COMPETING INTERESTS DISCLAIMER:**

Authors have declared that they have no known competing financial interests OR non-financial interests OR personal relationships that could have appeared to influence the work reported in this paper.

**DISCLAMIER ( ARTIFICIAL INTELLIEGENCE)**

Author(s) hereby declares that NO generative AI technologies such as Large Language Models (ChatGPT, COPILOT, etc.) and text-to-image generators have been used during the writing or editing of this manuscript.

References

1.) Google LLC, \*Data Isolation Strategies for SaaS in Big Query\*, Google Cloud Architecture Center, Accessed: Apr 2025.

<https://cloud.google.com/architecture/saas/data-isolation-strategies>

2.) Snowflake Inc., \*Snowflake Secure Data Sharing Documentation\*, Snowflake Documentation, Accessed: May 2025.

<https://docs.snowflake.com/en/user-guide/data-sharing-intro>

3.) Koreeda, T., Honda, H., & Onami, J.-i. (2025). Snowflake Data Warehouse for Large-Scale and Diverse Biological Data Management and Analysis. Genes, 16(1), 34. 2024 28th

International Conference on Information Technology (IT) <https://doi.org/10.3390/genes16010034>

4.) Katasani, D. P. (2025). Implementing Robust Data Governance Frameworks in the Snowflake Platform. International Journal of Scientific Research in Computer Science Engineering and Information Technology, 11(1).

https://doi.org/10.32628/CSEIT25111226

5.) Zhang, H., Liu, Y., & Yan, J. (2024). Cost-Intelligent Data Analytics in the Cloud. Proceedings of the Conference on Innovative Data Systems Research (CIDR)

https://doi.org/10.48550/arXiv.2308.09569

6.) Saiqa Aleem, Luiz F. Capretz, and Faheem Ahmed, “Security Issues in Data Warehouse: A Systematic Review,” Procedia Computer Science, 2015

https://doi.org/10.48550/arXiv.1507.05644

7.) Timo Schindler and Christoph Skornia, “Secure Parallel Processing of Big Data Using Order-Preserving Encryption on Google BigQuery,” arXiv preprint arXiv:1608.07981, 2016. https://arxiv.org/abs/1608.07981

8.) D. Seenivasan, “Optimizing Cloud Data Warehousing: A Deep Dive into Snowflake’s Architecture and Performance,” International Journal of Advanced Research in Engineering and Technology, vol. 12, no. 3, pp. 951–962, Mar. 2021 https://doi.org/10.2139/ssrn.5148190

9.) Google Cloud Team. (2025). Snowflake to Big Query Migration Guide.

<https://cloud.google.com/bigquery/docs/migration/snowflake-overview>

10.) Ali Gholami and Erwin Laure, “Security and Privacy of Sensitive Data in Cloud Computing: A Survey of Recent Developments,” arXiv, Jan. 2016

<https://doi.org/10.5121/csit.2015.51611>

11.) Snowflake Product Documentation. <https://www.snowflake.com/trending/data-centralization>

12.) Google Product Documentation. <https://cloud.google.com/bigquery-analytics-hub>

13.) https://cloud.google.com/bigquery/docs/best-practices-for-multi-tenant-workloads-on-bigquery#:~:text=Teams%20in%20tenant%20projects%20can,based%20on%20this%20recommended%20design

14.) <https://support.cordial.com/hc/en-us/articles/9615168271629-Snowflake-Secure-Data-Sharing#:~:text=With%20Secure%20Data%20Sharing%2C%20no,cluster%20shared%20data%20architecture>