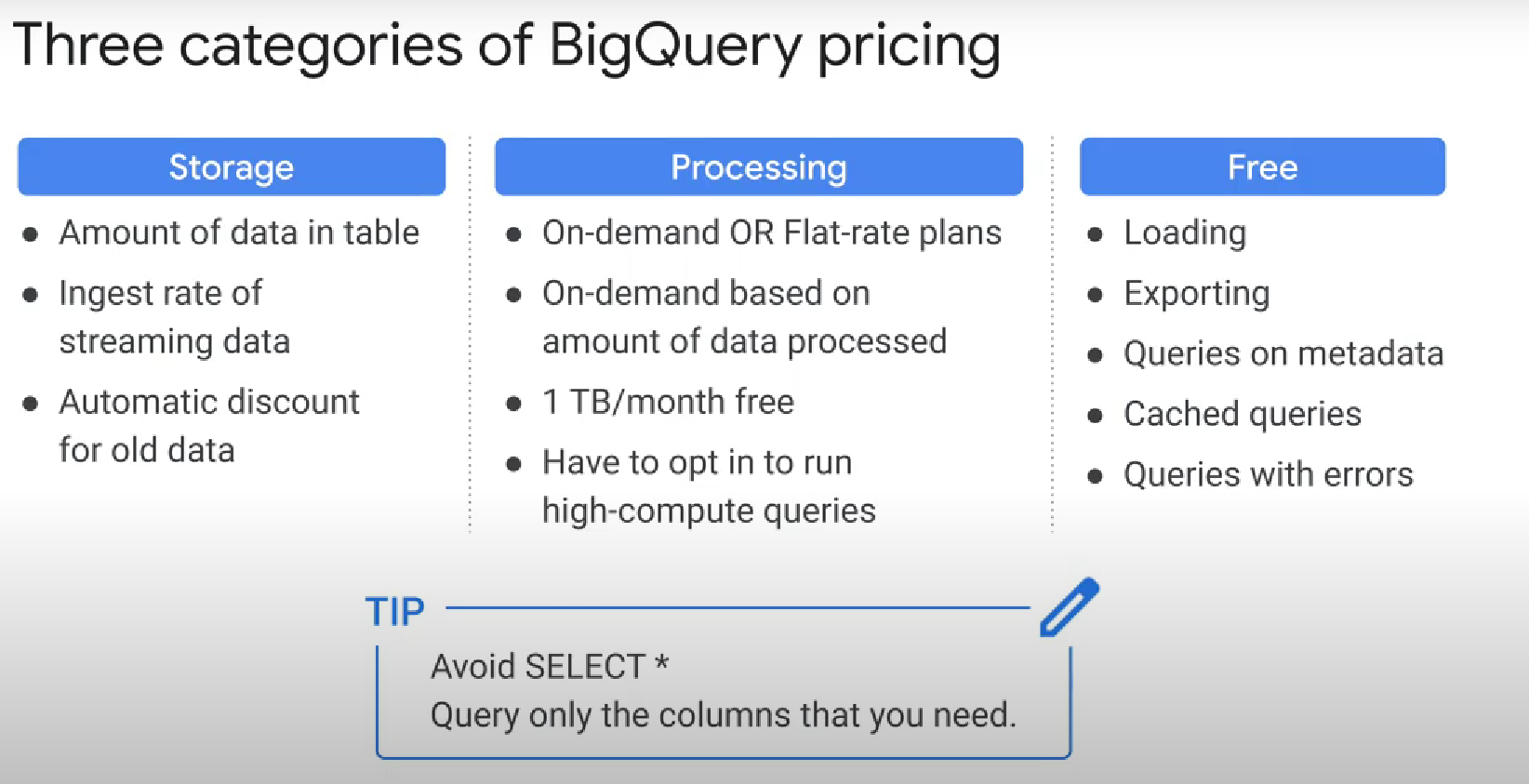


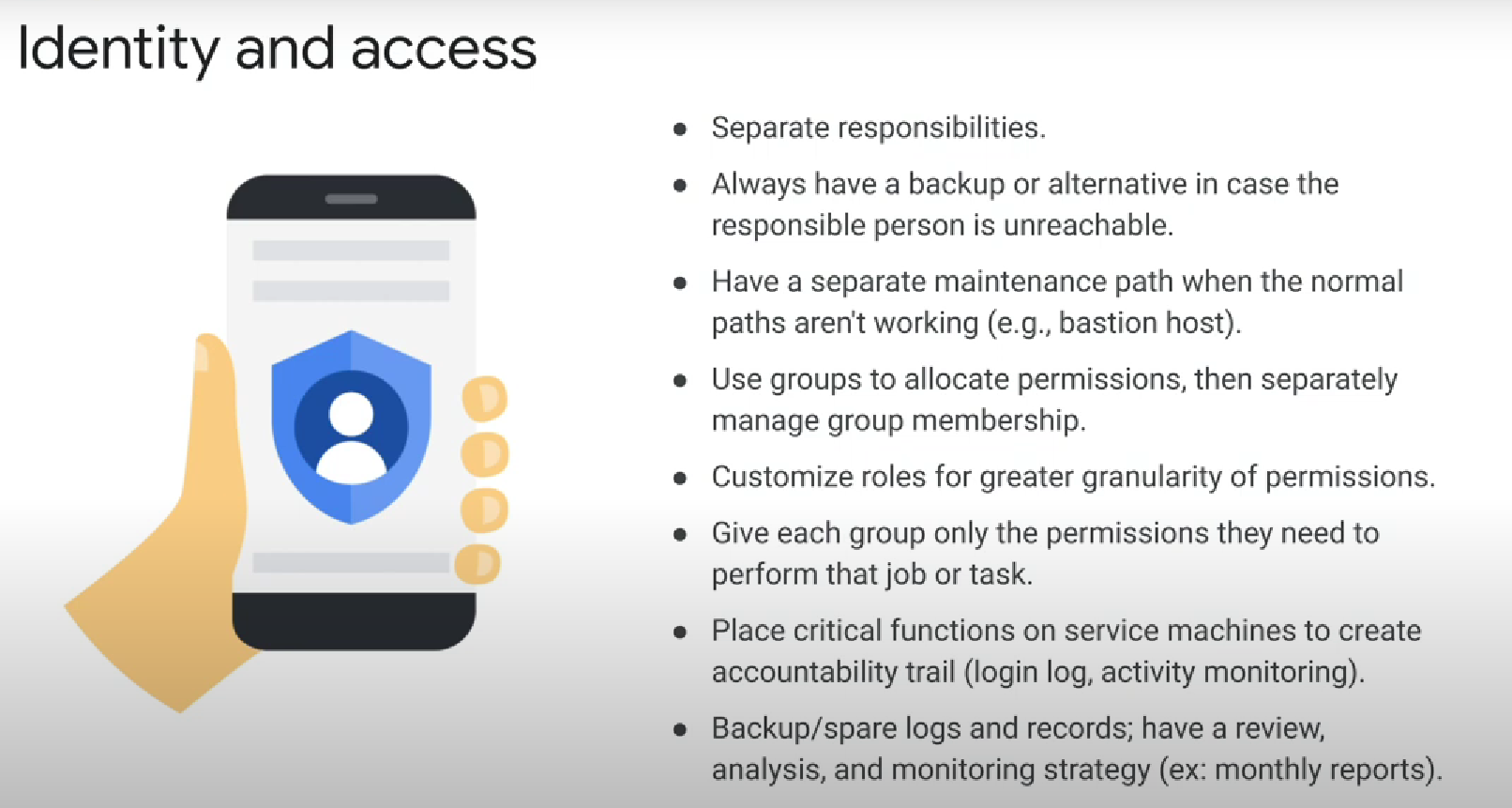
Google cloud pricing calculator can be used with BigQuery to estimate the cost of a query.

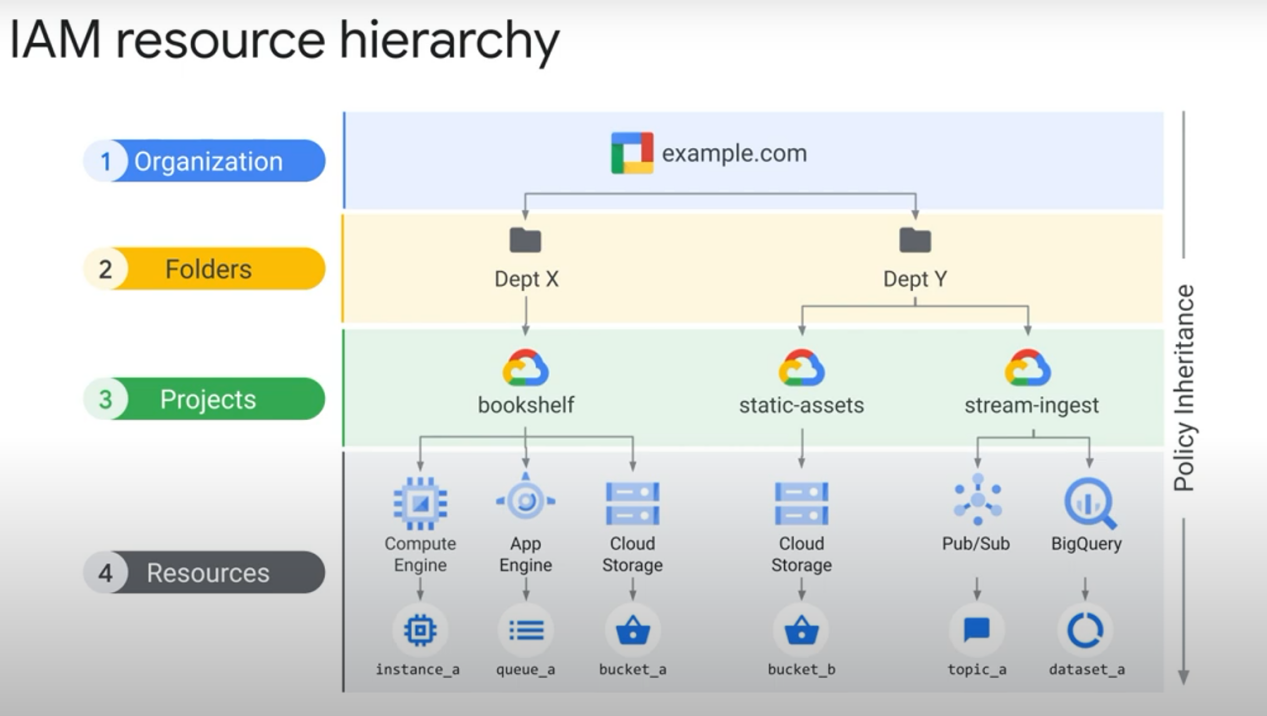
+

Query validator estimates about how much data will be used by the query. Use this data on the pricing calculator to get an idea of the cost



When you enroll in flat-rate pricing, you purchase dedicated query processing capacity, measured in BigQuery slots (cpu, memory, & network).

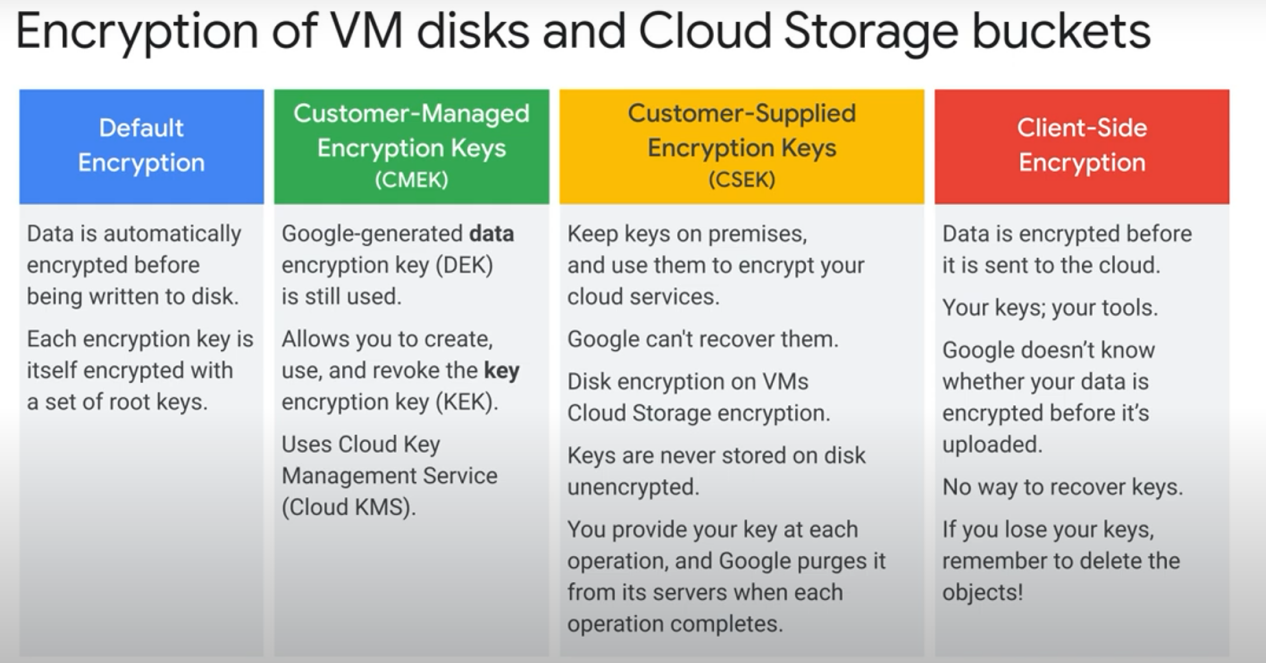




the final policy is the union of the parent policy and the resource policy what happens when these two policies are in conflict

what if the policy on the resource only gives access to a singlecloud storage bucket and restricts access to all other buckets however at the project level a rule exists that grants access to all buckets in the project

Answer: if the **parent policy is less restrictive** it **overrides** a more restrictive resource policy



key concepts

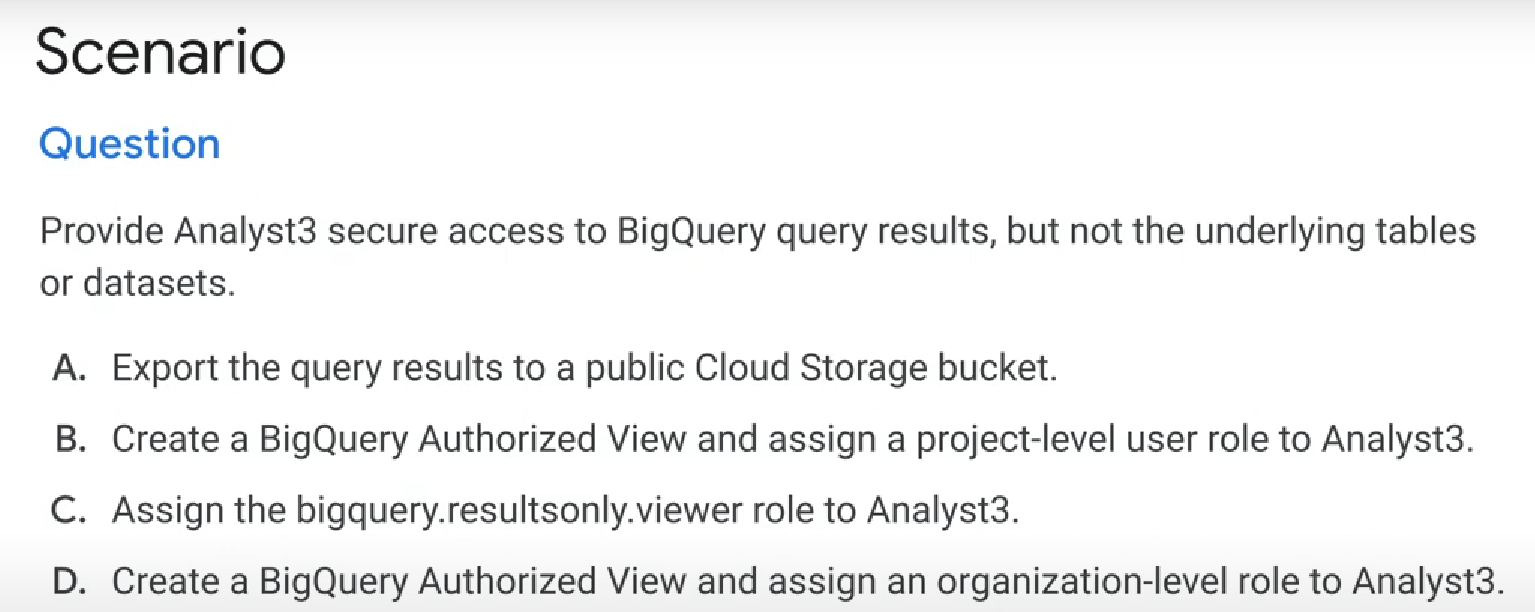
\* cloud armor: a service that protects from Ddos

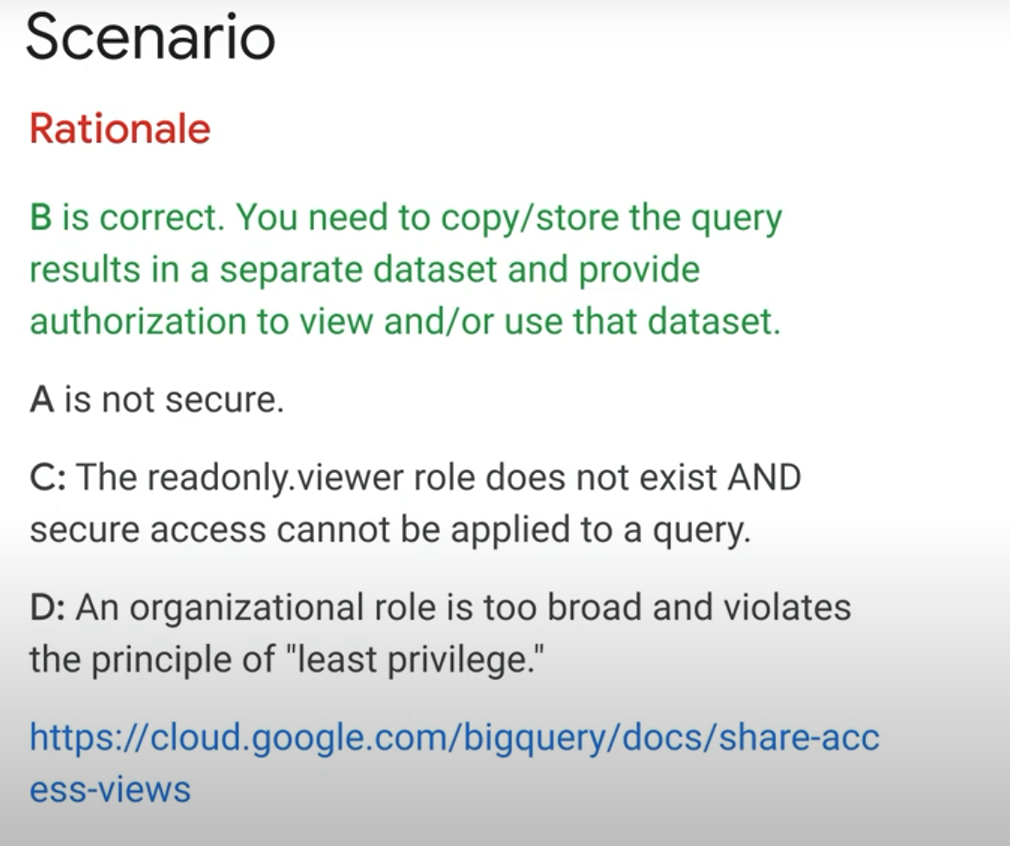
\* cloud load balancing

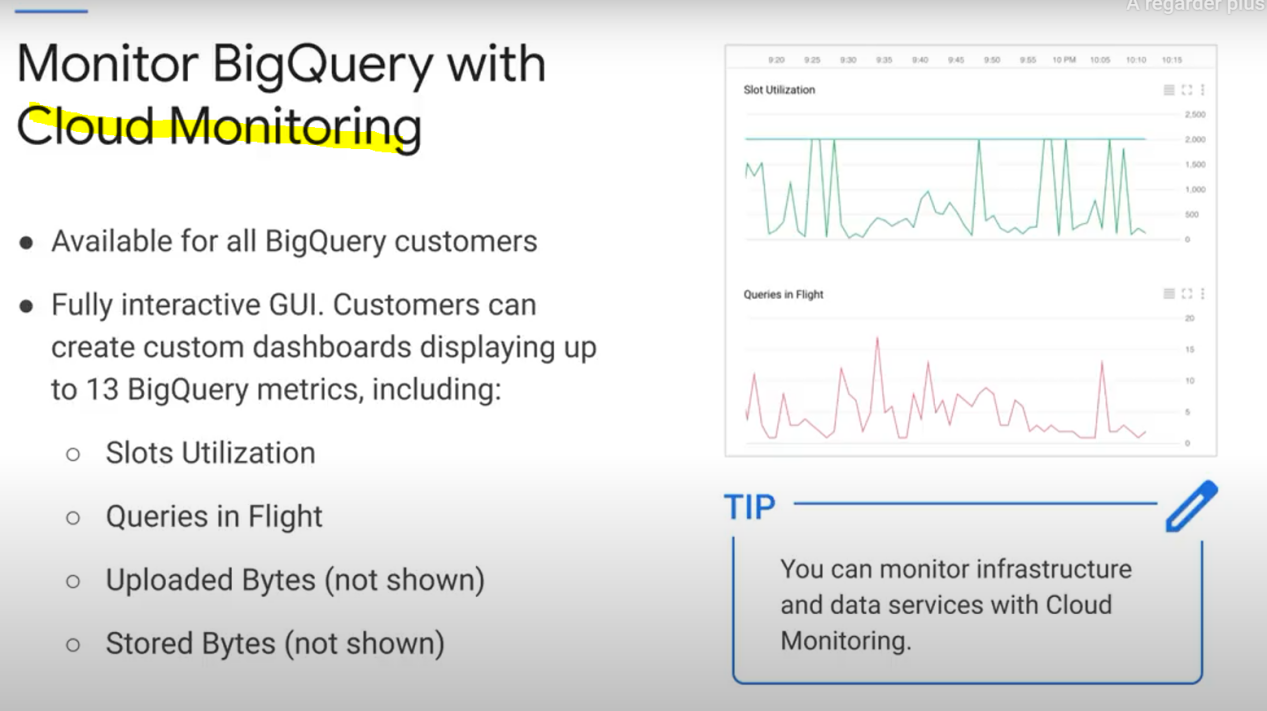
\* cloud firewall rules

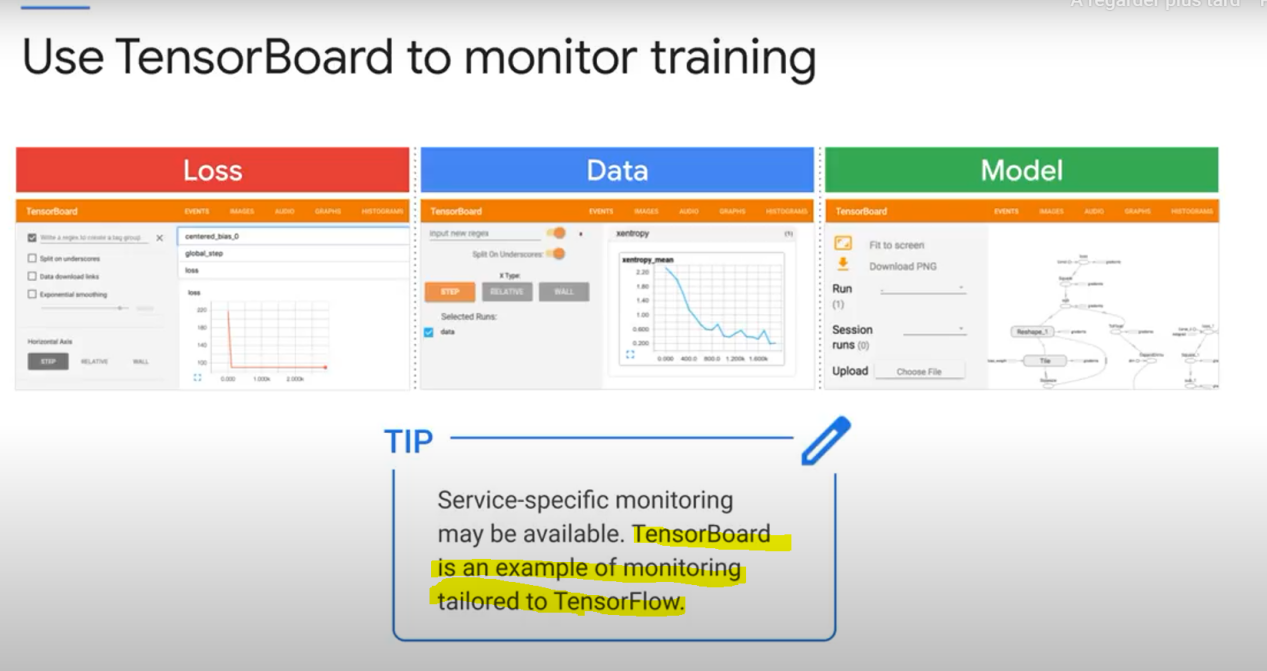
\* service accounts separation into front end and back end

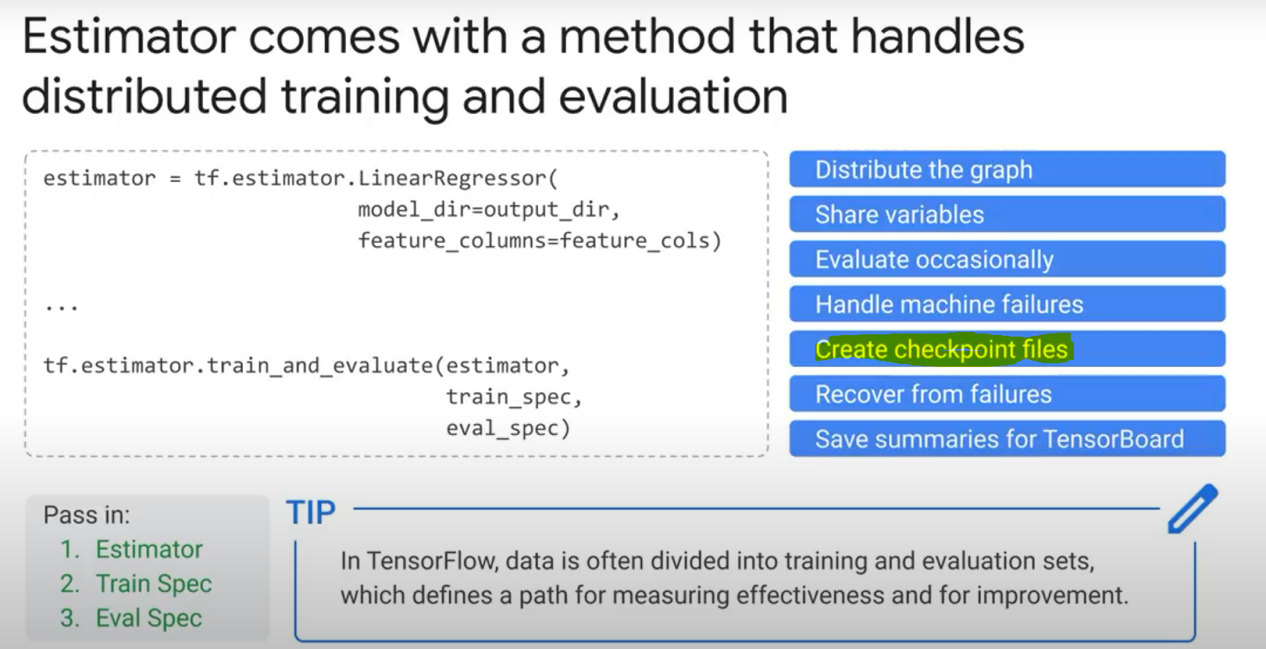
\* isolation of resources using separate service accounts between services











**Data protection (advocating policies):**

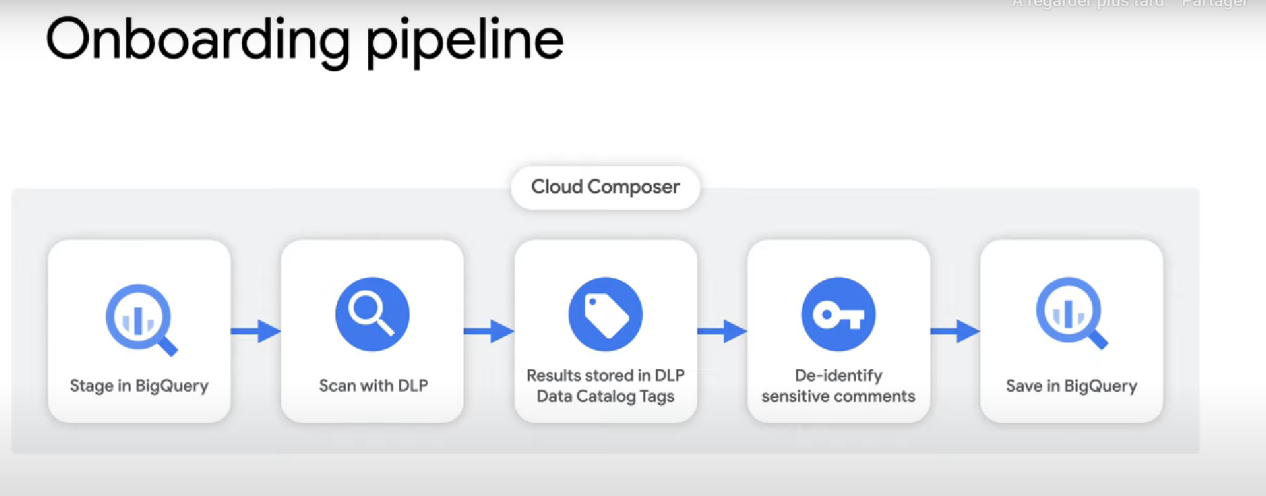
\* We can use **Data Catalog** to create a **Tag** that we will assign to certain column(s) containing personal information that we need to protect (pii)

Note: We can’t directly add a tag in bigquery to a specific column, as native bigquery tags are added to the entire table

You can use **«DLP» (cloud data loss prevention)** helps you discover sensitive data in datasets/tables.

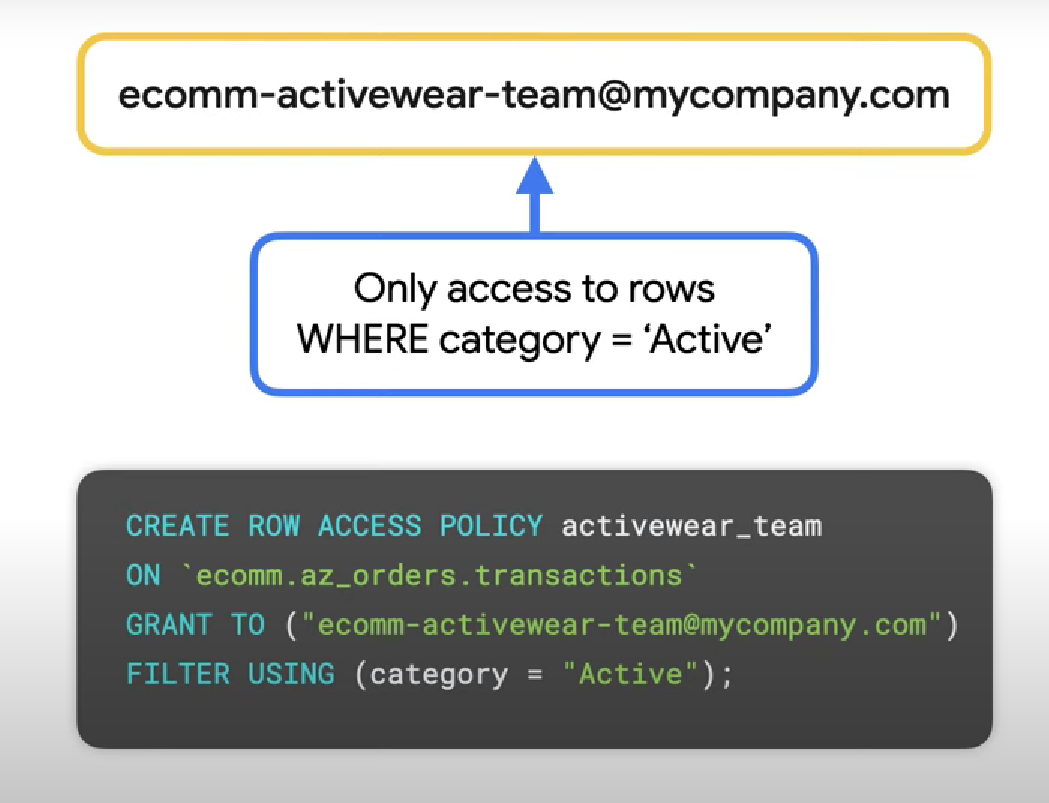
Official definition: Data security -- DLP -- Cloud DLP allows you to minimize what you collect, store, expose, or copy. Classify or automatically redact sensitive data from text streams before you write to disk, generate logs or perform analysis.

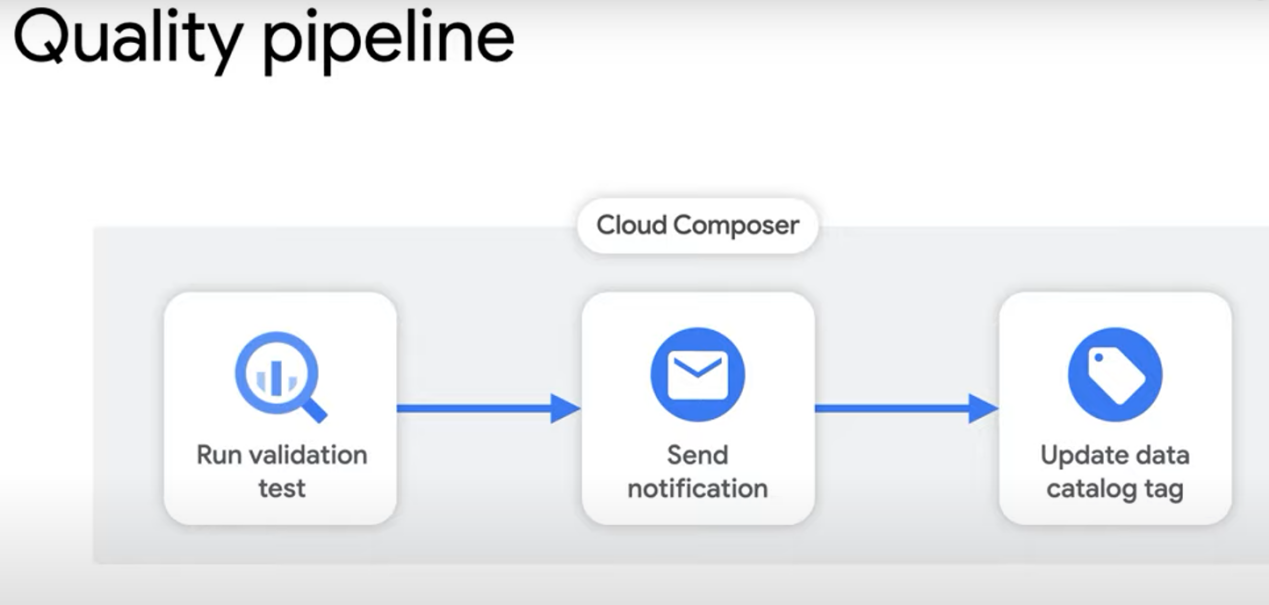
**De-identify** data: replace sensitive data with asterixes



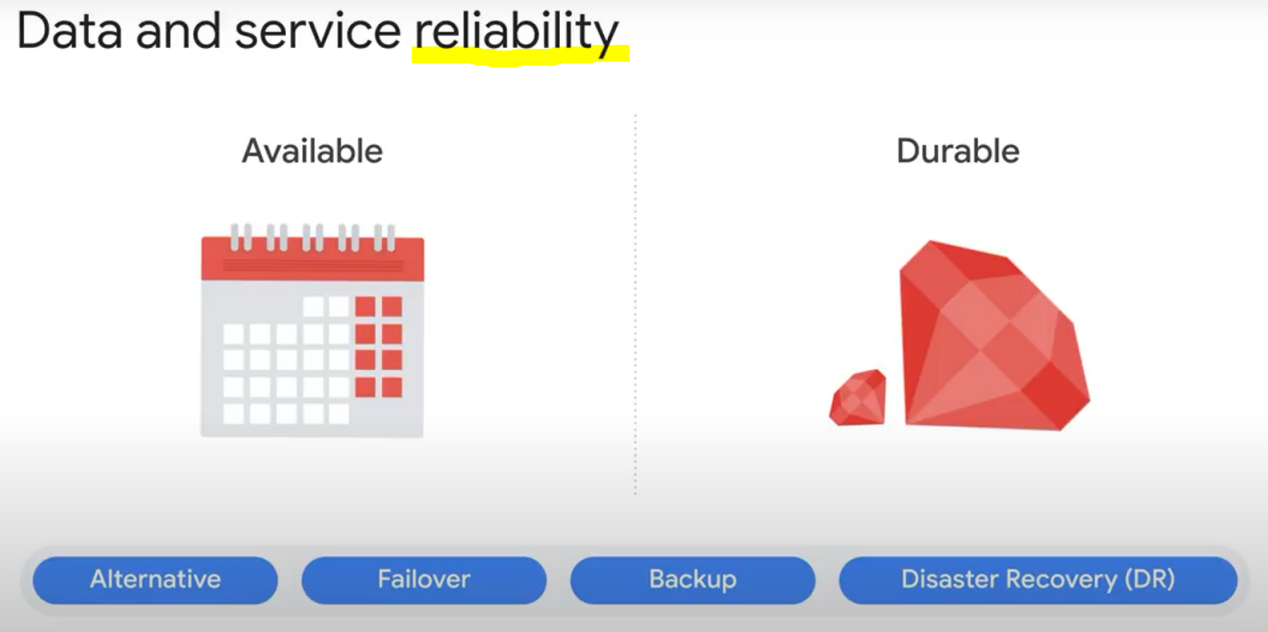
With data catalog, we can add a «clearance level» policy tag on specific column. It can have many values (E.g: low, medium, high). Then, any person who needs to read a column with a certain clearance (E.g: high), will need to be assigned the «fine-grained reader» role to the «High» ressource in data catalog.

You can also create a **row policy** that only allows certain readers to read rows with certain column/value pairs. E.g:





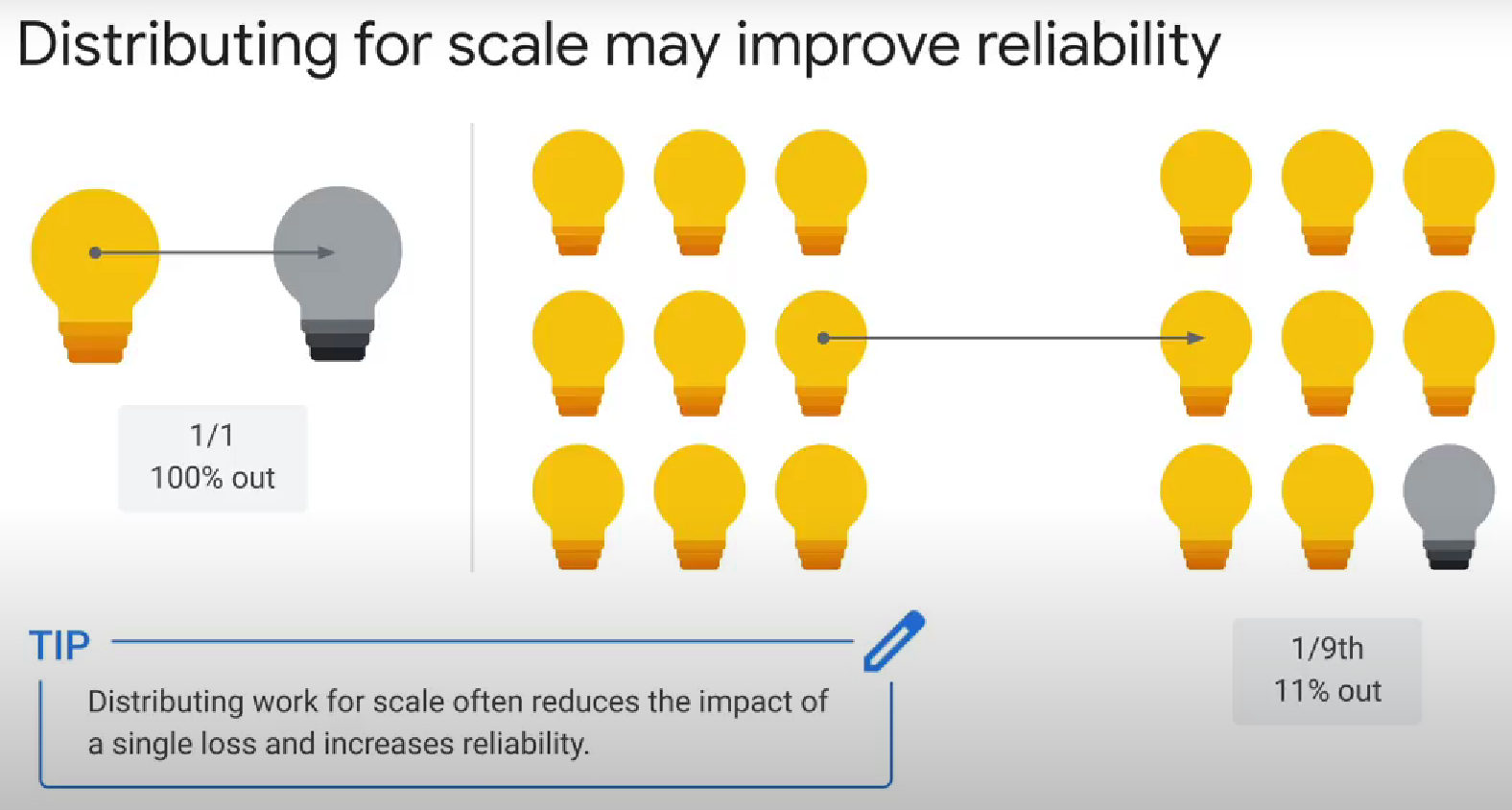
(data catalog is used here to add tags about data quality/validation to the processed data)



\* Automatic failover consists of automatically moving data or applications to the standby server if the primary system fails.

\* The alternative is a passive system where the process happens manually. Most failover processes operate automatically to reduce downtime.

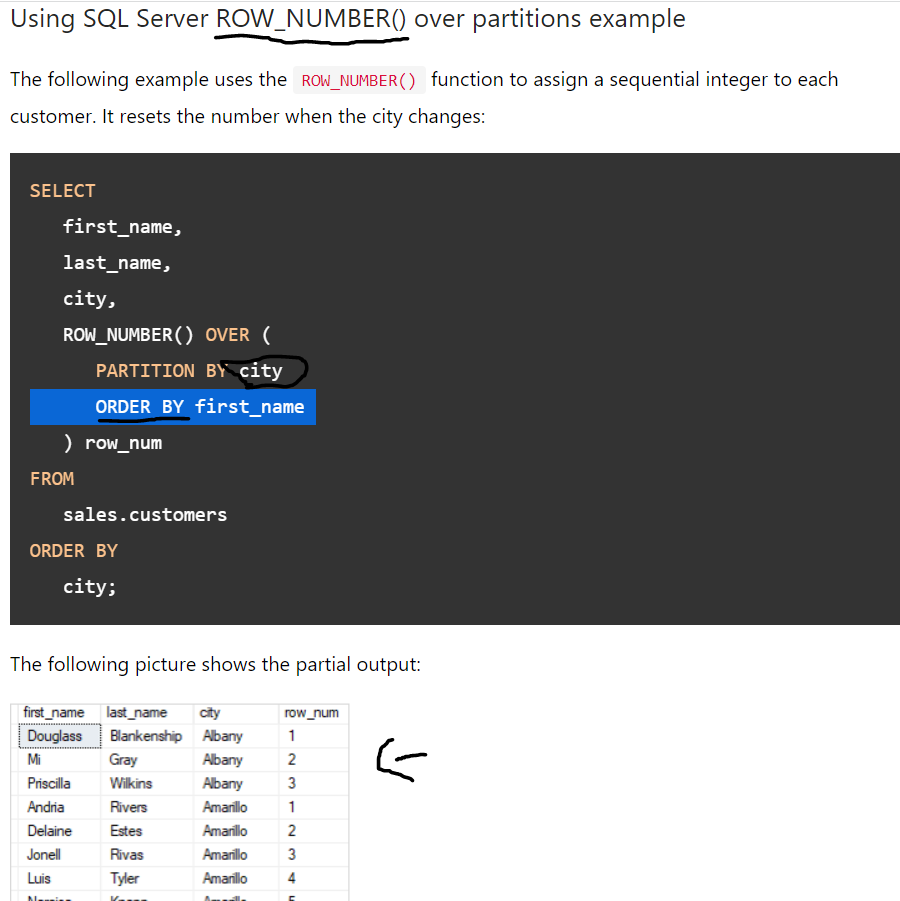
\* Backup and DR (disaster recovery) are the slowest. If we’re in a system that requires atomic transactions, we cannot accept these!



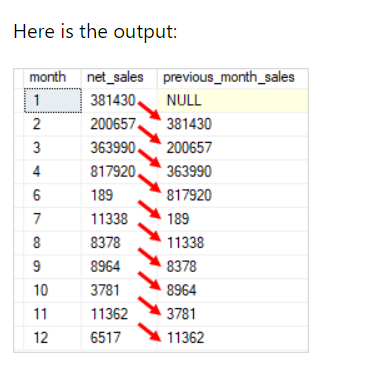
Google Cloud's operations suite = Cloud monitoring + Cloud Logging

Getting rid of duplicates when each row has an ID column, using a SQL query:

Use ROW\_NUMBER(Partition by ID, ORDER BY ID) as rowNumber, then add WHERE rowNumber = 1 (it starts at 1 not 0)

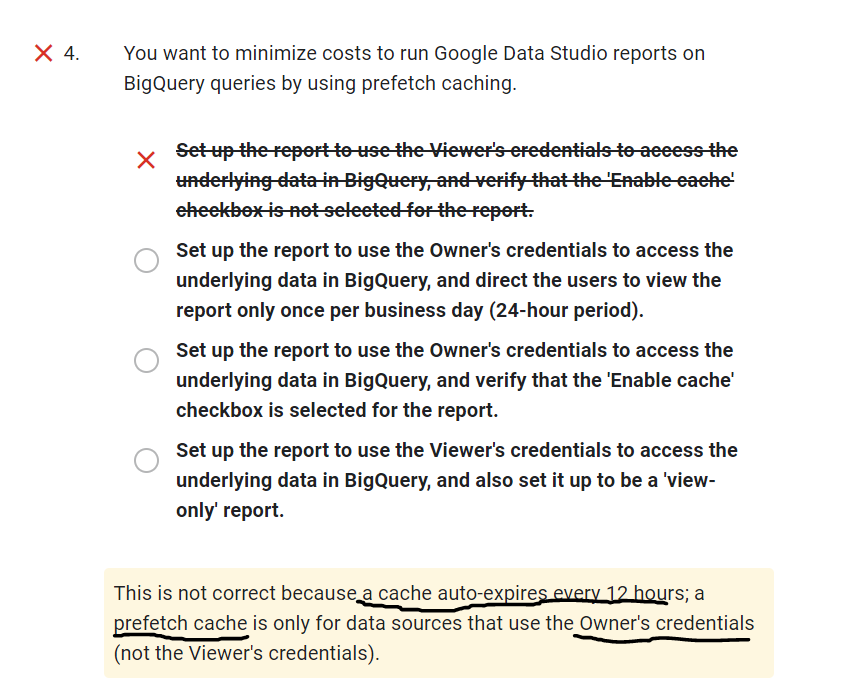


Lag function



(get 1 column from the nTH previous row (offset = n))

LEAD is the opposite of LAG



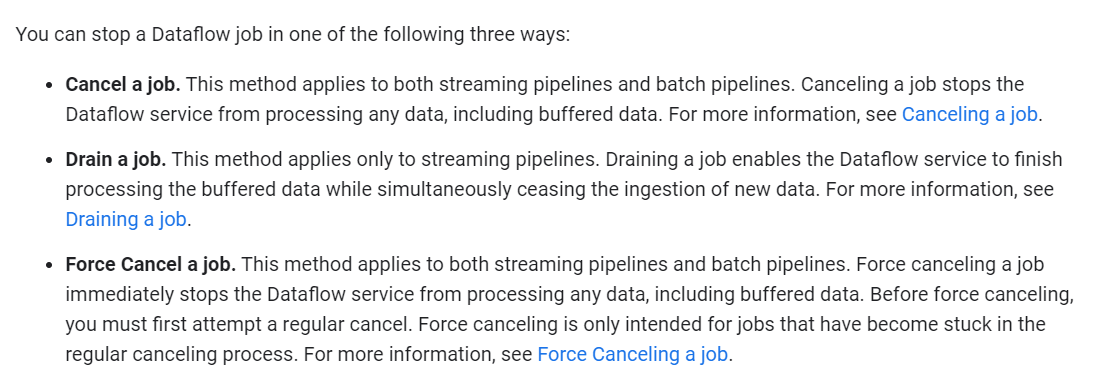
- **Avoid self-join at all cost** because that's what google says.

- **Adding additional compute resources is not a recommended way to resolve database schema problems.**

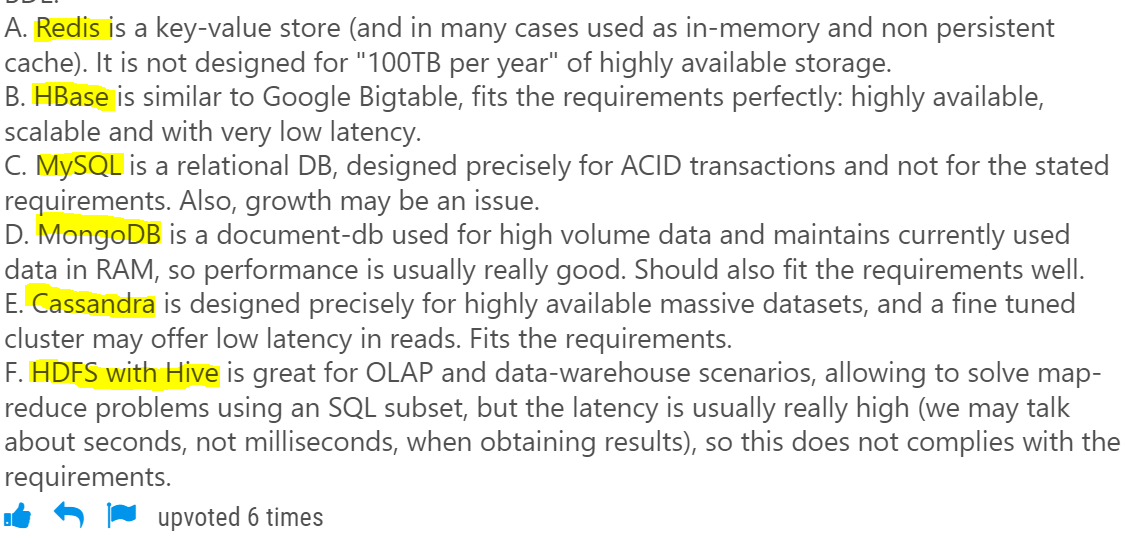
- For real-time data on Google Data Studio, disable caching by editing **the report settings (not bigquery’s)**

**- Cloud SQL** doesn’t auto-scale CPU power. It can only autoscale memory/storage.

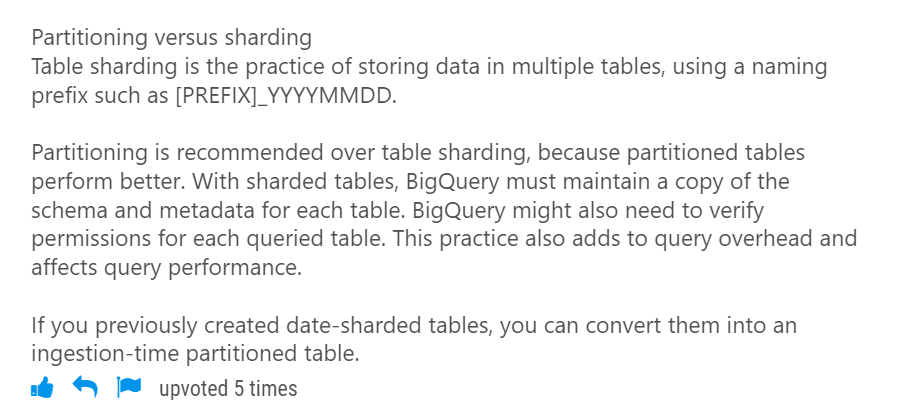
- **Google Cloud Datalab** is used for data vizualization and ML. You can deploy it in a VM on GCP.

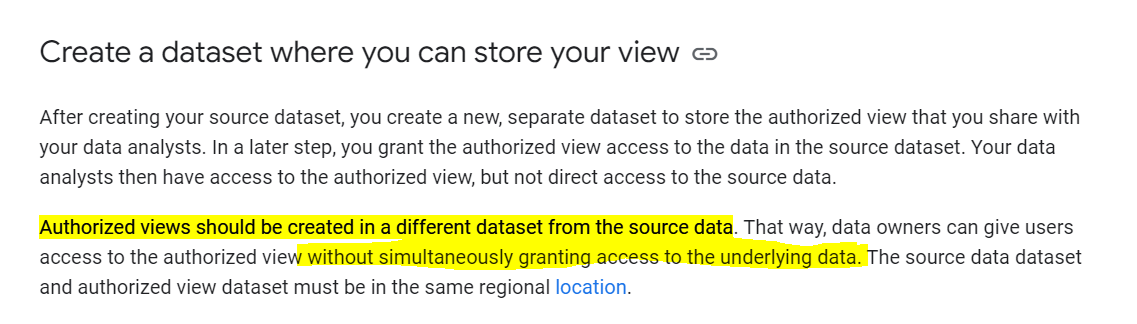
- 

Use the -drain flag when cancelling/replacing a dataflow job with another one to avoid losing data. Another option is to use the --update option, but that’s only if you’re not changing the windowing/triggers mechanism of your job (or just doing minor changes). With both options, buffered data won’t be lost.



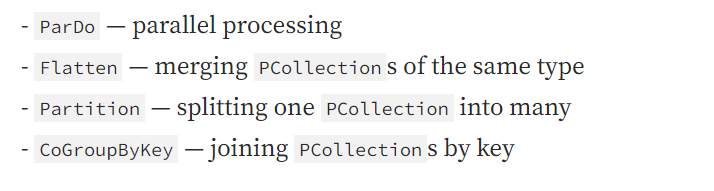
- Less complexity in ML model (less features, less layers, etc) + MORE DATA+ dropout & dropout layers + increasing regularization parameters (L1 & L2) = less risks of overfitting

- 



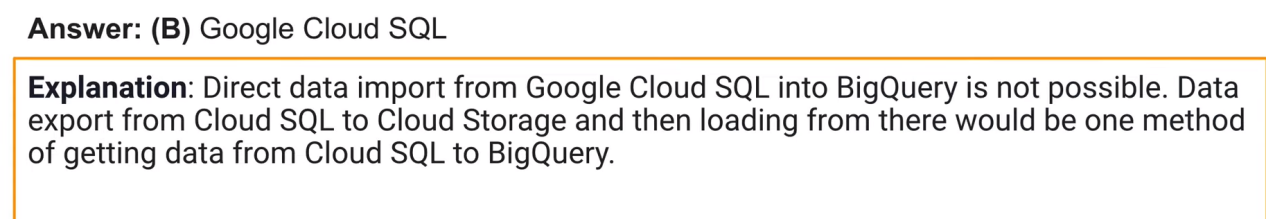
- **BigQuery Reservations** enable you to switch between on-demand pricing and flat-rate pricing

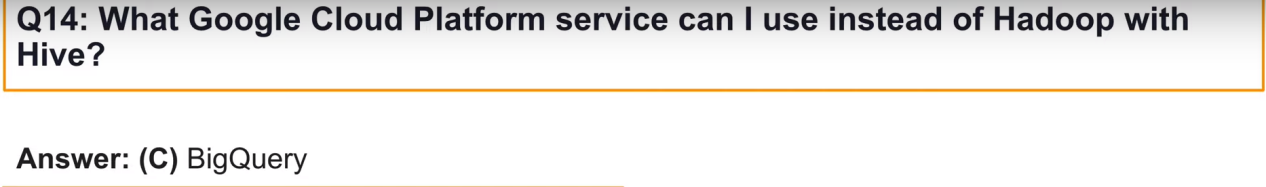
After you purchase slots, you can assign them to different buckets, called reservations. Reservations let you allocate the slots in ways that make sense for your particular organization. Jobs that require high SLAs will take priority to use the reserved slots.

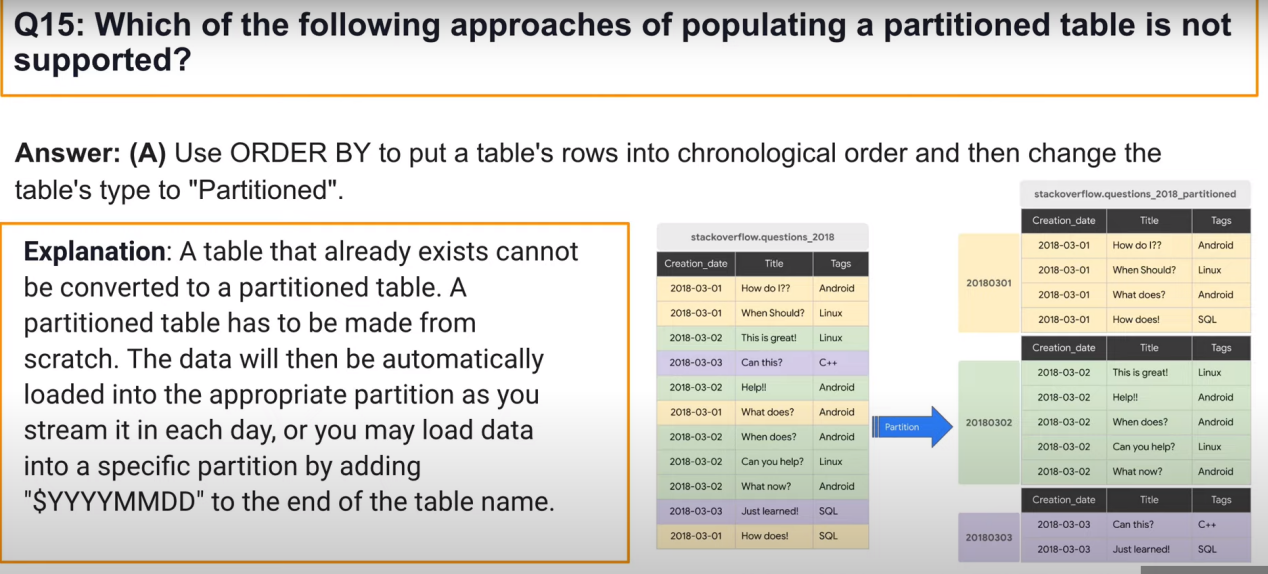


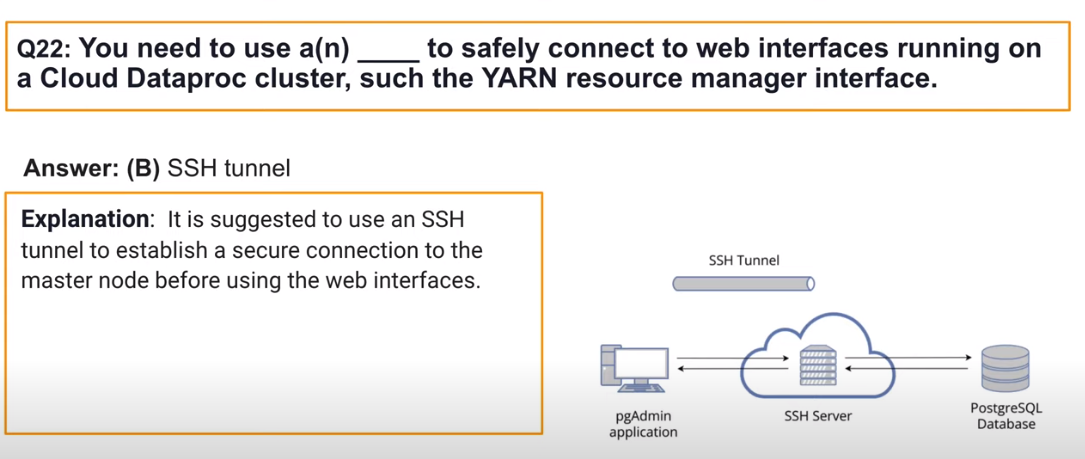
-In Event time the data is processed based on the timestamp at the source of each record. It is essentially the time in which event is created.

Process time is the time of receival of data at the streaming application. This is also the time where the data is processed at the streaming service.

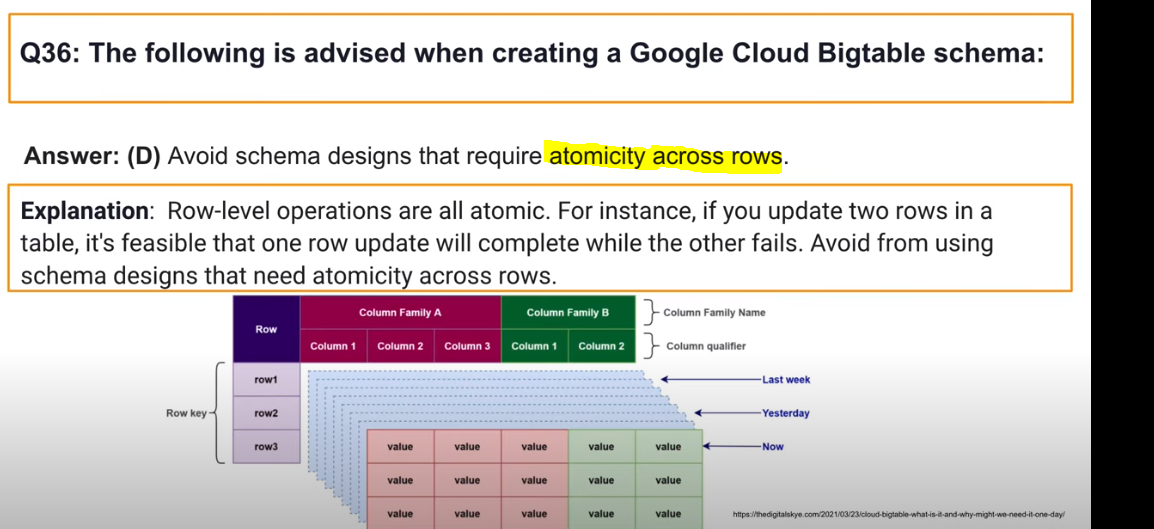
-





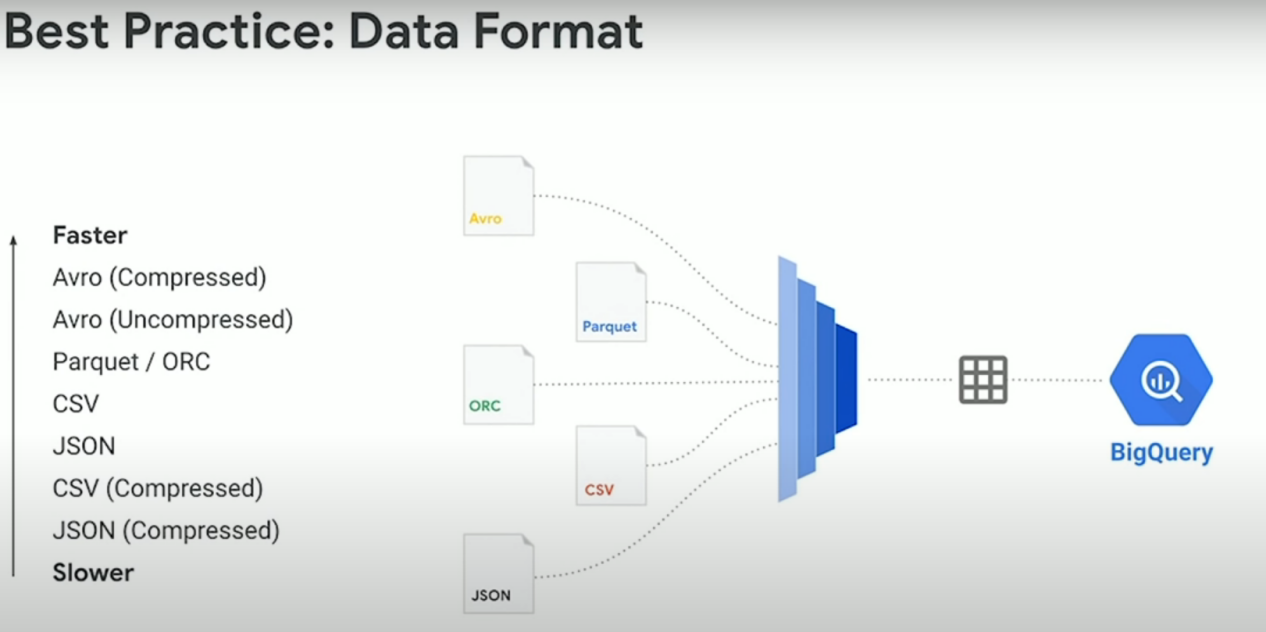


- Bigtable: A row key must be **4 KB or less (KEEP IT SHORT)**. Long row keys take up additional memory and storage and increase the time it takes to get responses from the Bigtable server



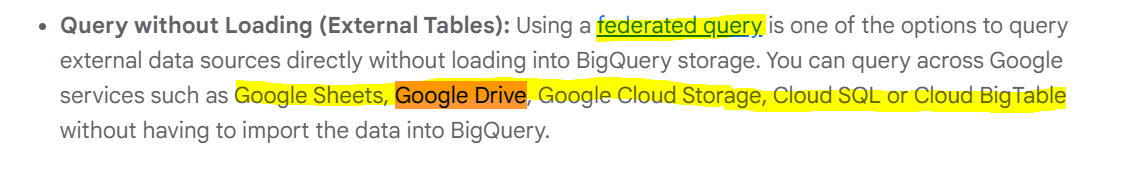
Solving hotspotting: **Field promotion:** involves moving fields from the column data into the row key to make writes non-contiguous. For example, TIMESLIDERS could promote the USERID from a column to an element of the row key.

If it didn’t work, salting: create a hash from your timestamp, divide it by the number of nodes you haves, append the remainder to the row key = + simple - Adds complexity

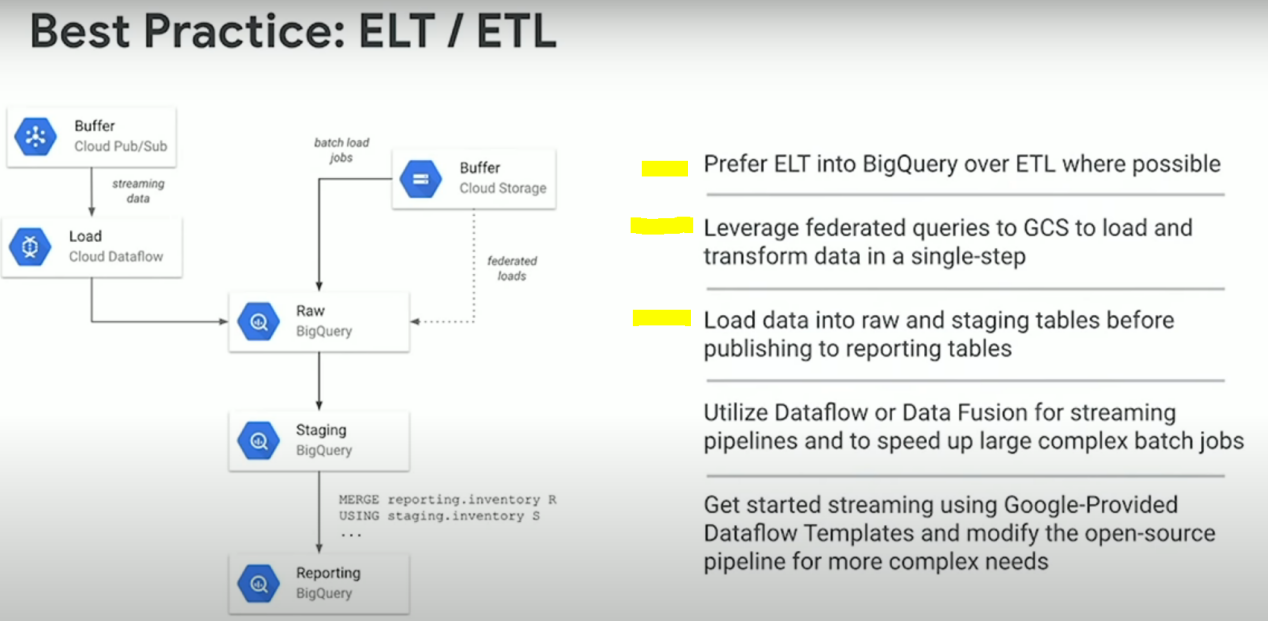


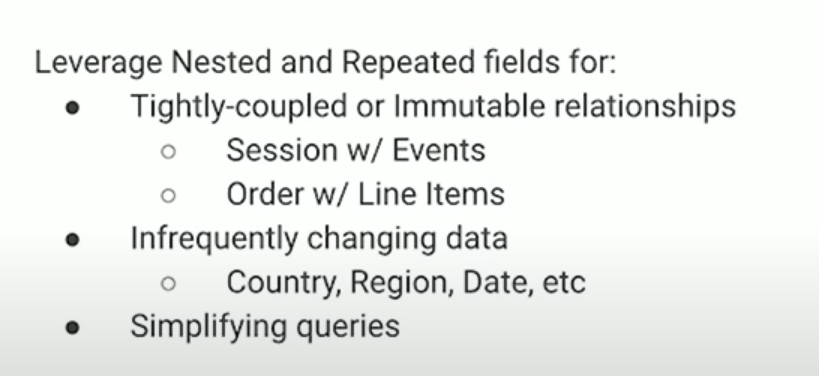
(you can also load data to bigquery from

(you can also query data outside of bigquery:

)

(faster dans le sens load job speed)



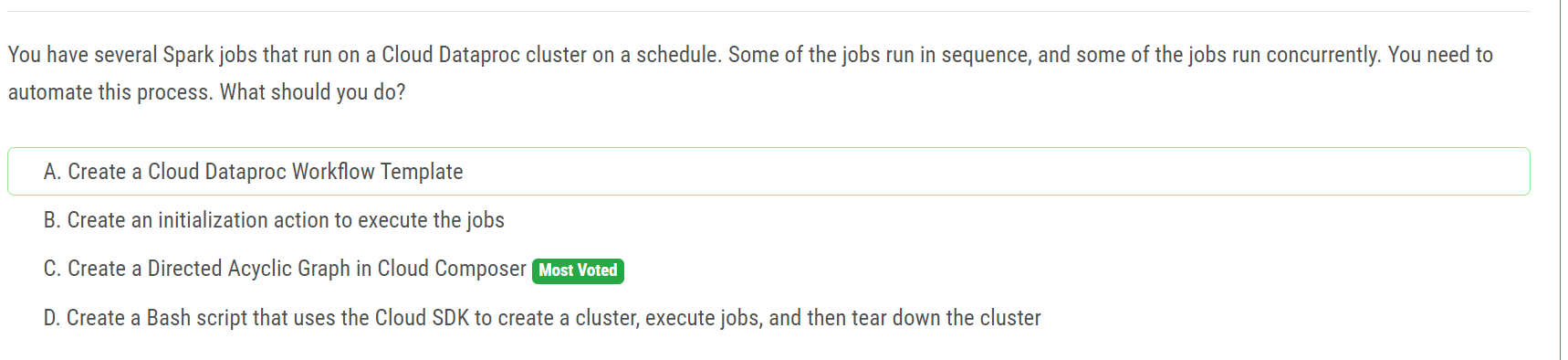


\*Cloud composer can schedule queries

- You can export audit logs to bigquery to be able to analyze them. You can also visualize them on google data studio.

\*Keep a record for longer than seven days. With BigQuery time travel, you can only access a table's data from seven days ago or more recently. With table snapshots, you can preserve a table's data from a specified point in time for as long as you want.

Minimize storage cost. BigQuery only stores bytes that are different between a snapshot and its base table, so a table snapshot typically uses less storage than a full copy of the table.

-

C is correct, because workflow templates don’t allow you to schedule jobs

-Sinks control how Cloud Logging routes logs. Using sinks, you can route some or all of your logs to supported destinations (pub/sub, cloud storage, bigquery, another cloud logging bucket)

Sinks belong to a given Google Cloud resource: Cloud projects, billing accounts, folders, and organizations. When the resource receives a log entry, it routes the log entry according to the sinks contained by that resource. The log entry is sent to the destination associated with each matching sink.

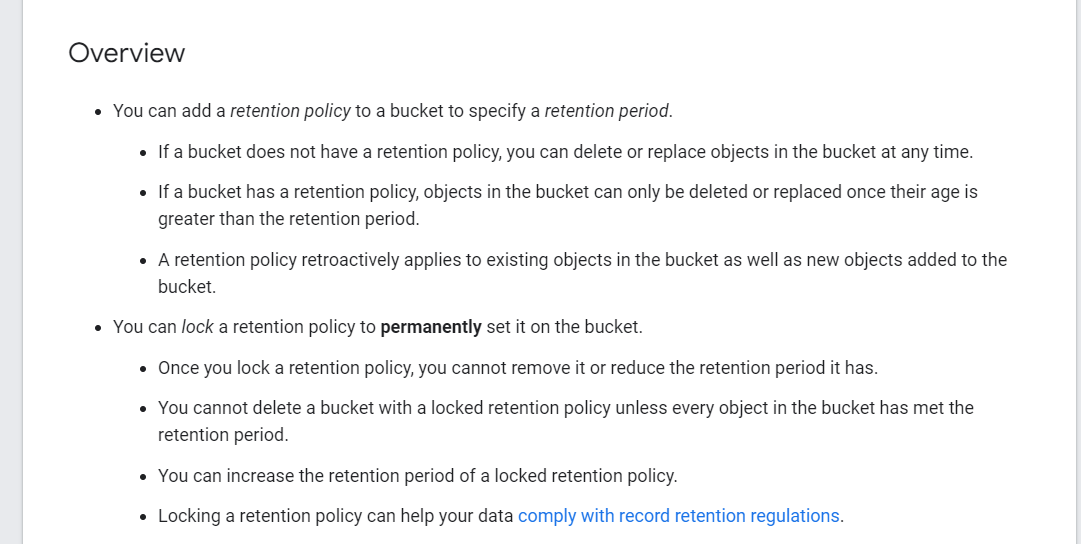
**An aggregated sink** is a type of sink that combines and routes log entries from **all the Google Cloud resources contained by an organization or folder.**

-All interdependent tasks (cannot run in parallel, run one after the other) need to be run through **cloud composer** whereas small/adhoc tasks need to be run via **cloud scheduler**

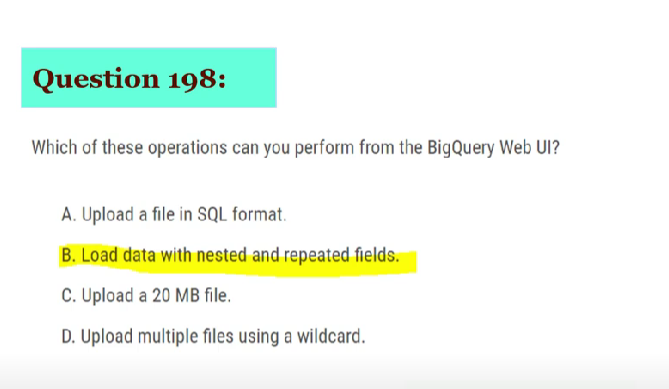
For could SQL:

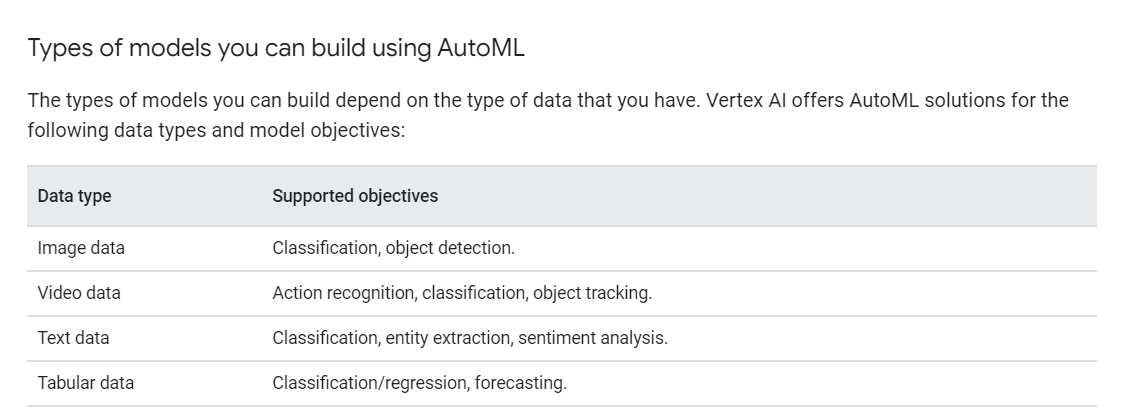
\*Read Replicas CAN be promoted to master nodes in the case of disaster recovery (DR). However, there is downtime entailed.

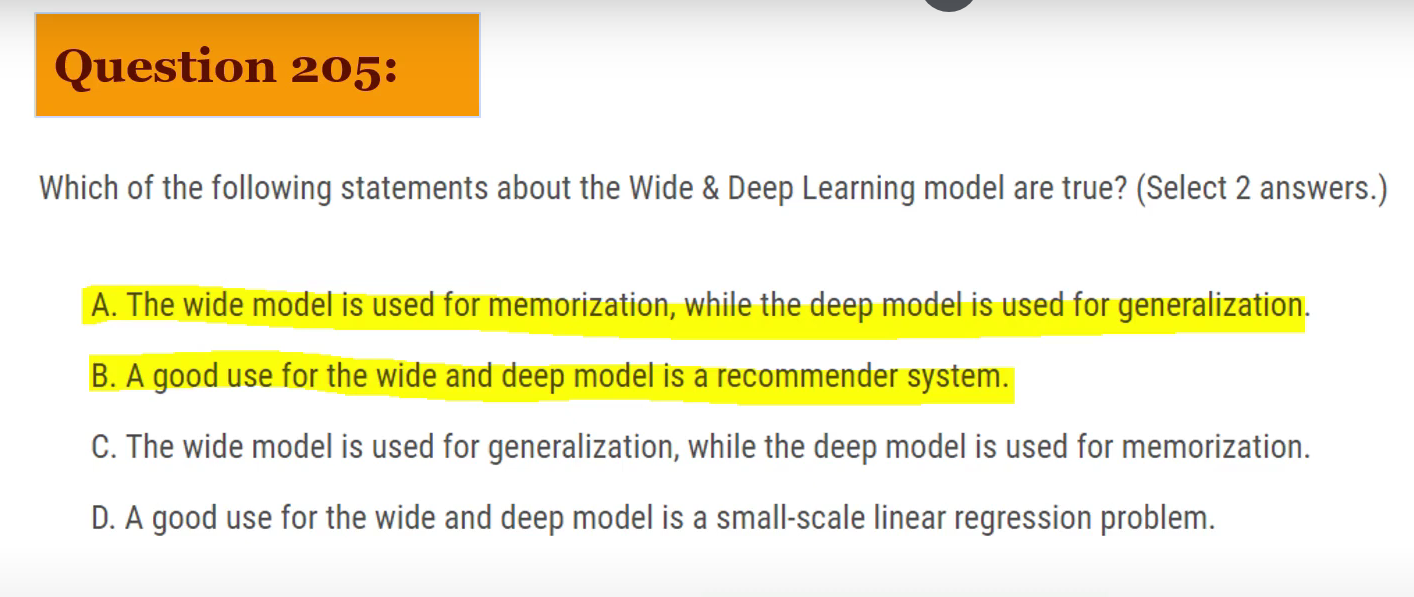
\* Failover Replicas are designed to automatically become master nodes.



-By default, BigQuery caches query results for 24 hours, sauf si table protected by column-level security

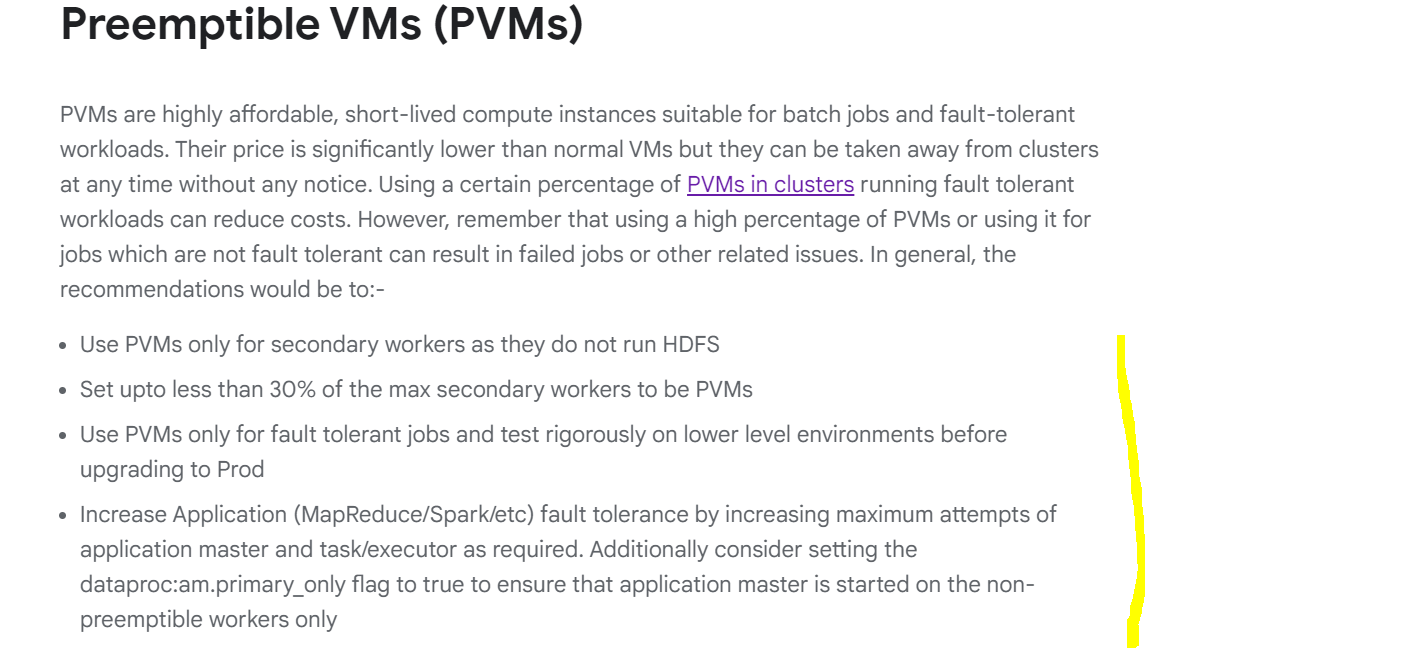




**Nee**

-(Dataproc) Because of graceful decomission which needs to be set to a value longer than the longest job you have on your clusters: for sensitive long running workloads, consider scheduling on separate **ephemeral clusters**. Conversely, keep short running jobs on a single auto scaling cluster.

-



-compute intensive use cases can benefit from more vCPUs (compute optimized machines [C2]) while allocating more memory persistent disks for i/o intensive ones

- For monitoring, to create custom dashboard = export your metrics to stackdriver. To create alerts, find saturated ressources, and general monitoring = Cloud Monitoring (Cloud monitoring is the new version of stackdriver I think)

-Set dataproc:job.history.to-gcs.enabled to true to minimize local disk consumption. (will store mapreduce and spark jobs history to GCS instead of HDFS = less risk of saturation of disk space)

- The **Spark history** **server and YARN history server UI** is useful to view and debug corresponding applications.

Use the **diagnose utility** to obtain a **tarball** which can provide a snapshot of the cluster’s state at the time (contains a snapshot with EVERYTHING)

- With a clustered table, a LIMIT clause can reduce the number of bytes scanned, because scanning stops when enough blocks are scanned to get the result. You are billed for only the bytes that are scanned.

- In general, Bigtable offers optimal latency when the CPU load for a cluster is under 70%. For latency-sensitive applications, however, we recommend that you plan at least 2x capacity for your application's max Bigtable queries per second (QPS).

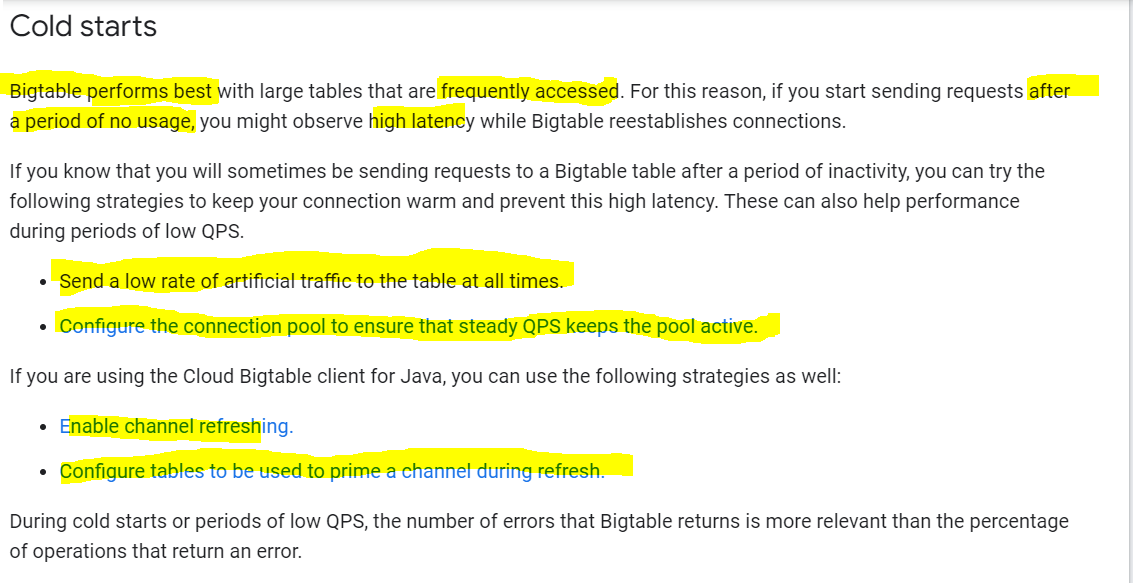
- (BigTable) For latency-sensitive applications we recommend that you keep storage utilization per node below 60%. If your dataset grows, add more nodes to maintain low latency.

&

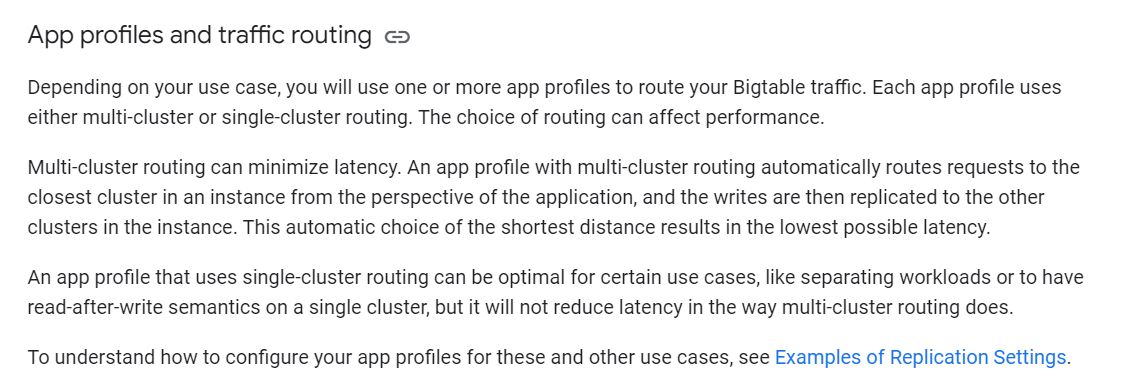
Optimal max row size = 1KB & Avoid having too many columns (slows the number of rows per second)

&

In an instance that uses **replication**, each cluster must handle the work of replication in addition to the load it receives from applications. Underprovisioned clusters can cause increased latency.



**Bigtable data replication:**

1. Replication can improve **read throughput**, especially when you use multi-cluster routing. Additionally, replication can reduce read latency by placing your Bigtable data geographically closer to your application's users.
2. Although replication can improve availability and read performance, **it does not increase write throughput.**
3. It takes some time for changes to propagate for a cluster to the replicas (the other clusters)
4. **Node Usage:** As explained in Write throughput, when an instance uses replication, each cluster in the instance must handle the work of replication in addition to the load it receives from applications. For this reason, a cluster in a multi-cluster instance often needs more nodes than a cluster in a single-cluster instance with similar traffic.
5. 

Multi-cluster routing: if there’s a problem with the main cluster, automatically failover to the next nearest cluster

Single-cluster routing: no automatic failover, you need to do it manually (good for budget, you’ll be taking the decision to move the processing to other regions (which may cost more))

-Bigtable Prevents you from enabling single-row transactions in an app profile that uses multi-cluster routing, because there's no safe way to enable both of these features at once. (because Bigtable only ensures single-row transactions)

**to-study if I have time:**

-Exam questions: One was about GKE and containers and the most efficient way to deploy the container and the infrastructure. The other was ML model with PARTIAL custom C++ TF ops - to choose accelerator.

- HIPPA, COPPA, FedRAMP, GDPR policies (data protection)

- Partitionning VS clustering in bigquery?

Clustering is the same as partionning, but you can use any column type (text, number, etc...), not only timestamps/datetime/number like in partionning. Also, note that your table can have a maximum of 4 clustered columns per table, and that the order of the columns (which column we cluster first) is very important for query performance: always cluster the most queried/filtered by column first.

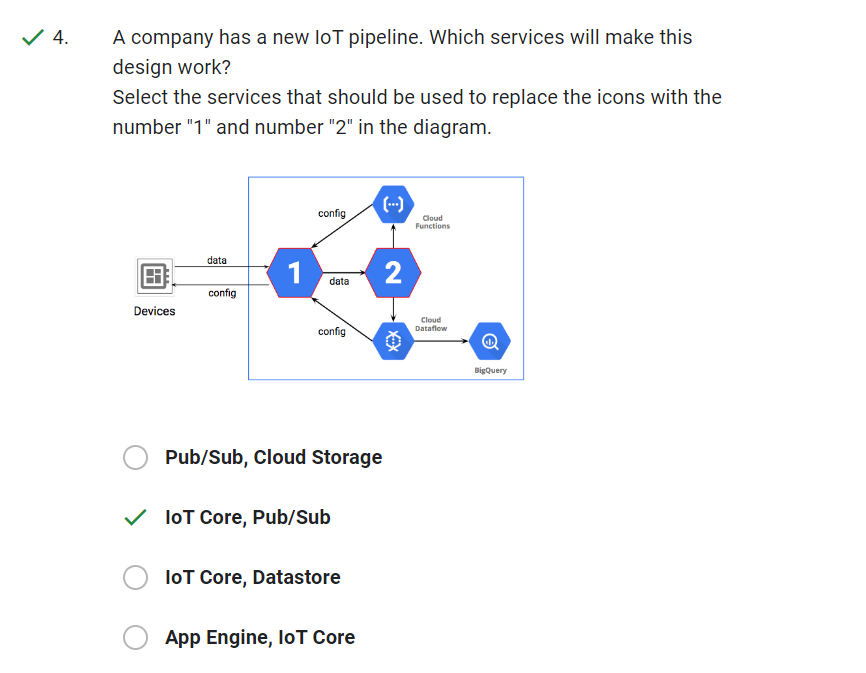
- partition expiration dates?

When you create a table partitioned by ingestion time or time-unit column, you can specify a partition expiration. This setting specifies how long BigQuery keeps the data in each partition.

- TODO: read about SIGNED URLS

- TODO: LEARN ABOUT THE IAM ROLES

- IOT Core :



- Cloud Router with VPN:

\* Cloud VPN (public internet), VS direct peering VS Partner/Dedicated interconnect

Note: VPC Network Peering is used for data transfers within Google Cloud Organizations.

Note 2: Cloud VPN is useful for data transfers at a rate of a few Gbps (1.5 Gbps to 3 Gbps).

\* Cloud VPN securely connects your peer network to your Virtual Private Cloud (VPC) network through an IPsec VPN connection. Traffic traveling between the two networks is encrypted by one VPN gateway and then decrypted by the other VPN gateway. This action protects your data as it travels over the internet. You can also connect two instances of Cloud VPN to each other.

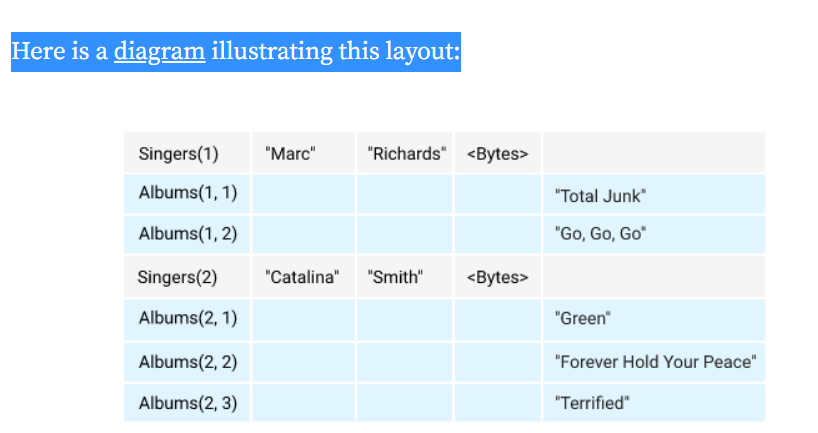
- Partner VS direct interconnect:

**Dedicated Interconnect** provides a direct physical connection between your on-premises network and Google's network. **Partner Interconnect** provides connectivity between your on-premises and VPC networks through a supported service provider.

Note: If your traffic doesn't require a 10-Gbps or 100-Gbps circuit, consider **Cloud VPN** or **Partner Interconnect** instead of a dedicated interconnect.

Note 2: For partner interconnect: more points of connectivity through one of our supported service providers. Traffic flows between networks through a service provider, **not through the public internet.**

-Cloud Spanner, interleaving tables to optimize join queries:

Interleaved tables: 1 table is the «parent», the other is the «child», and they are stored in the same place. Therefore, the join queries you use on the 2 tables will be very optimized

-Hopping Windows = Sliding

- Tumbling Windows = Fixed