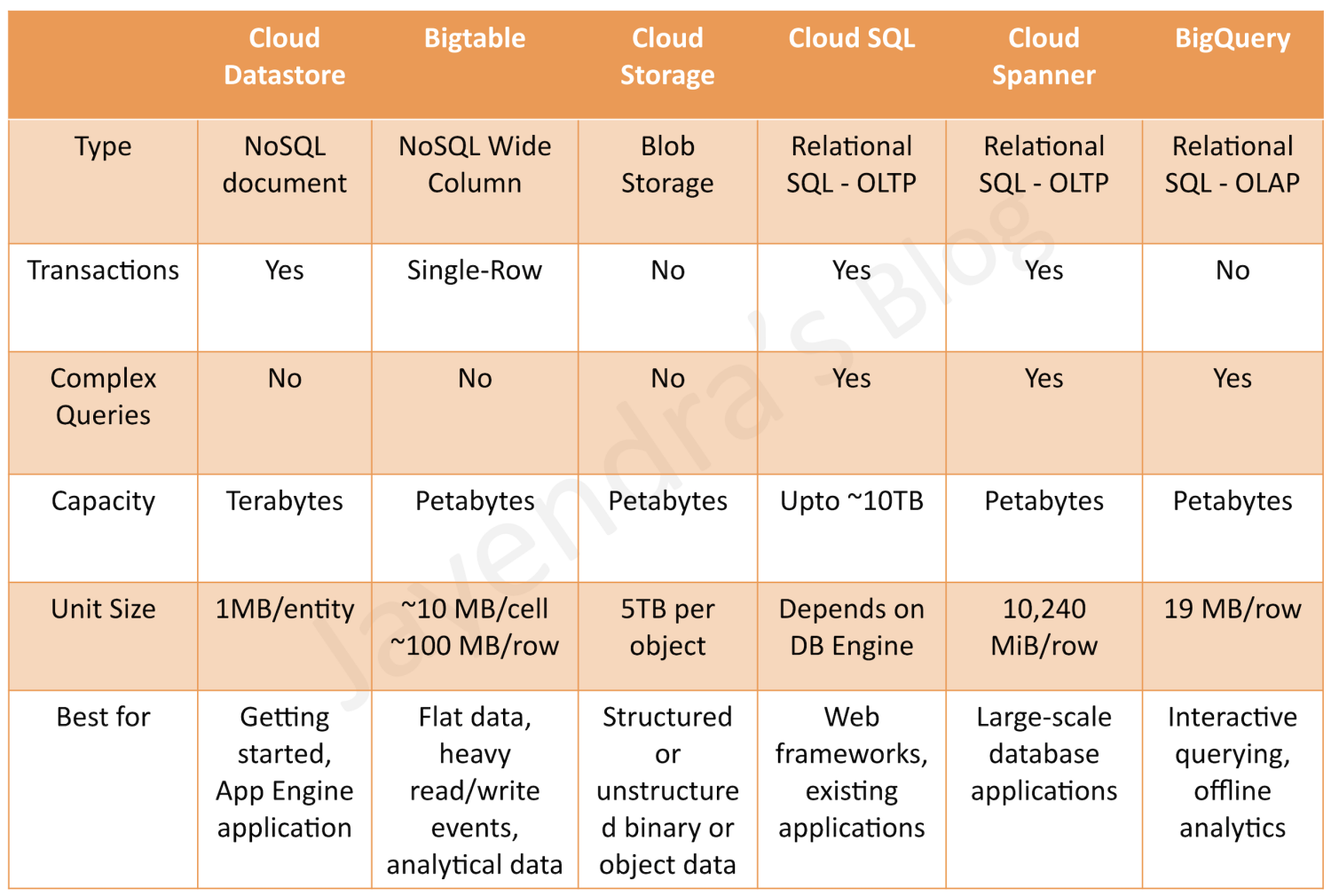
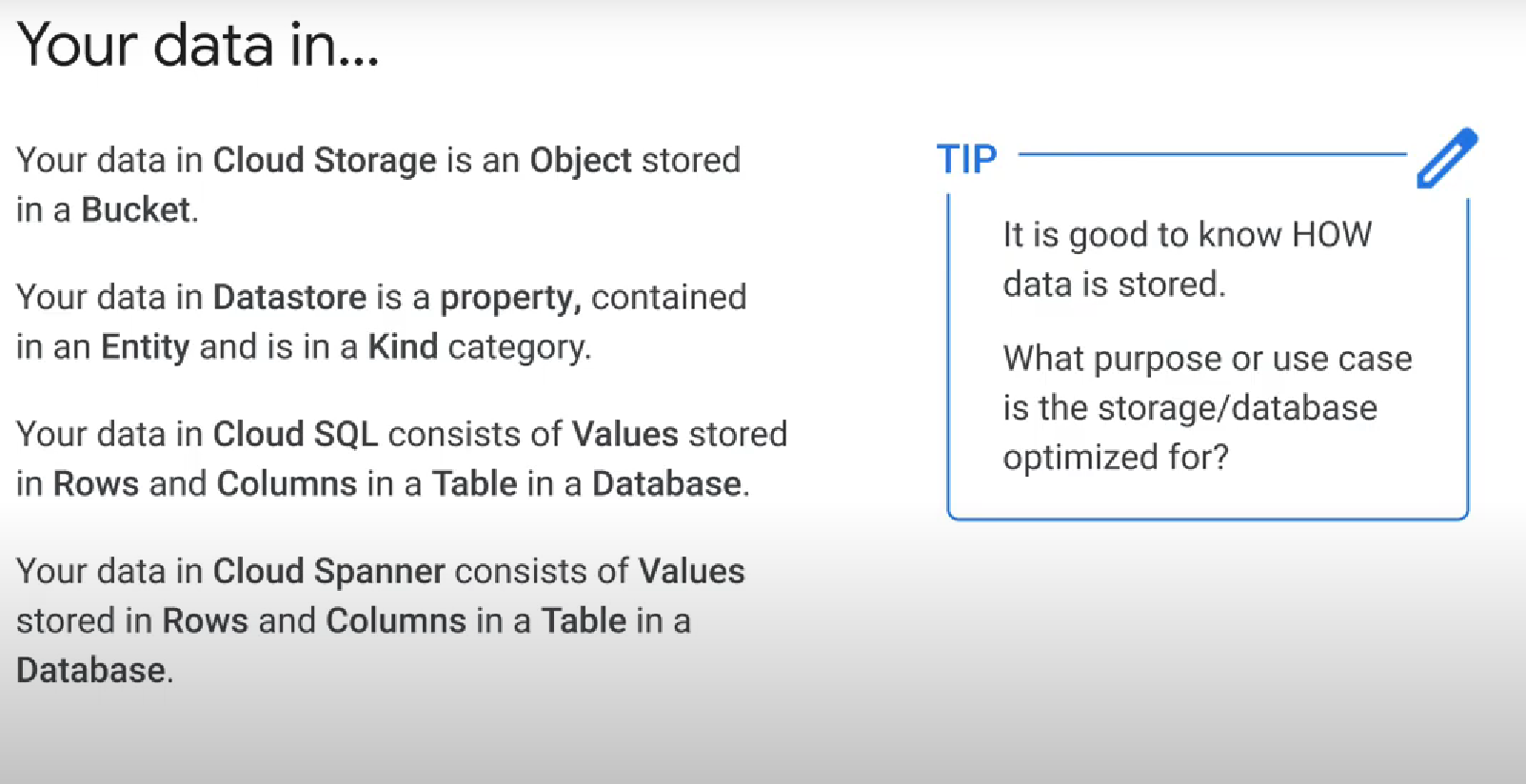
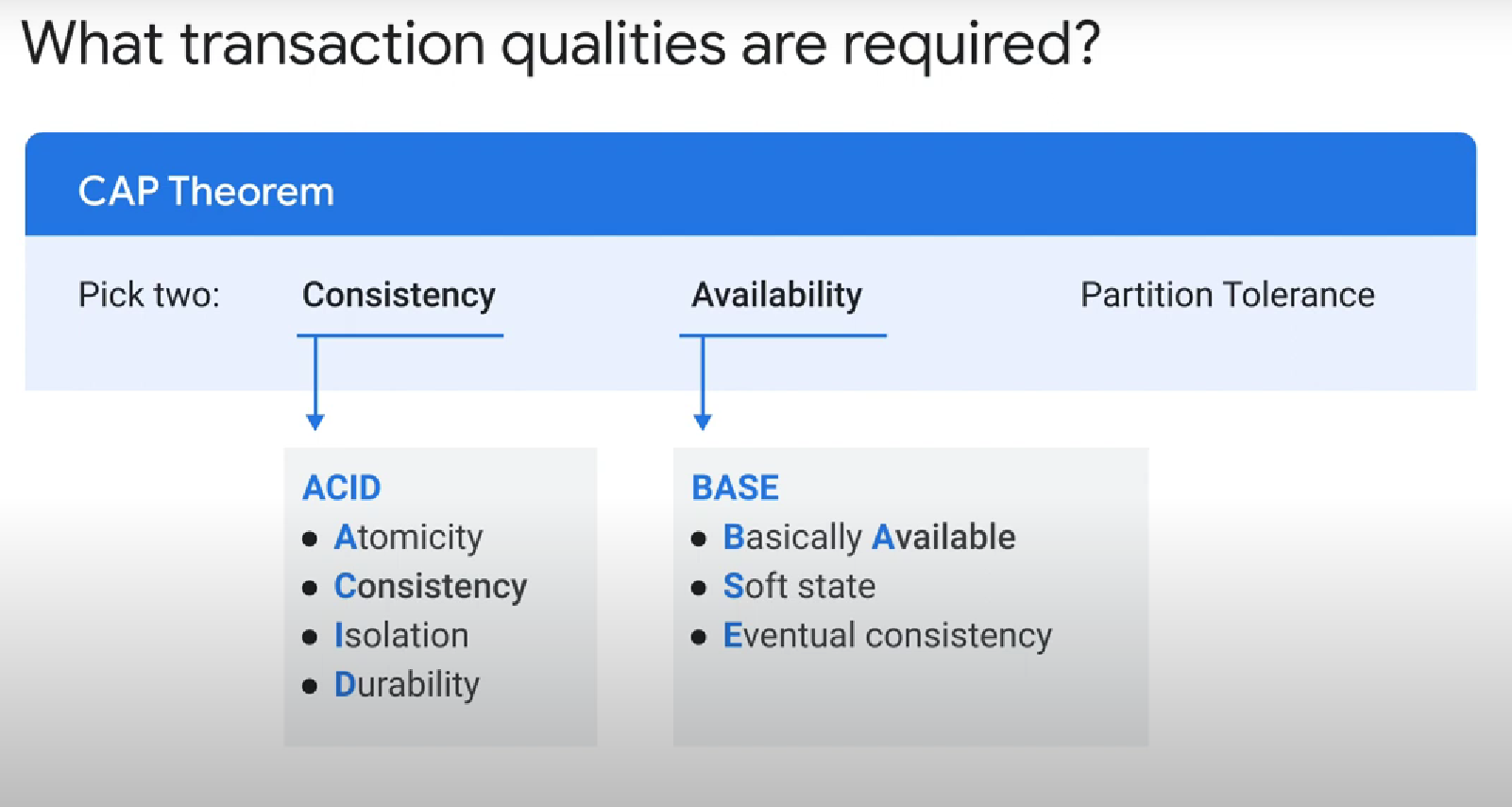


**For streaming data, why use bigtable instead of cloud Spanner? COST**

**E.g: for 100,000 Queries per second, bigtable needs 10 nodes VS 150 nodes in Spanner**





**Cloud Spanner Vs BigQuery**: https://hevodata.com/learn/spanner-vs-bigquery/#1

Main points:

- BigQuery is better for data analysis

- Cloud spanner supports transactions

- Cloud spanner offers TrueTime, a very realiable global clock

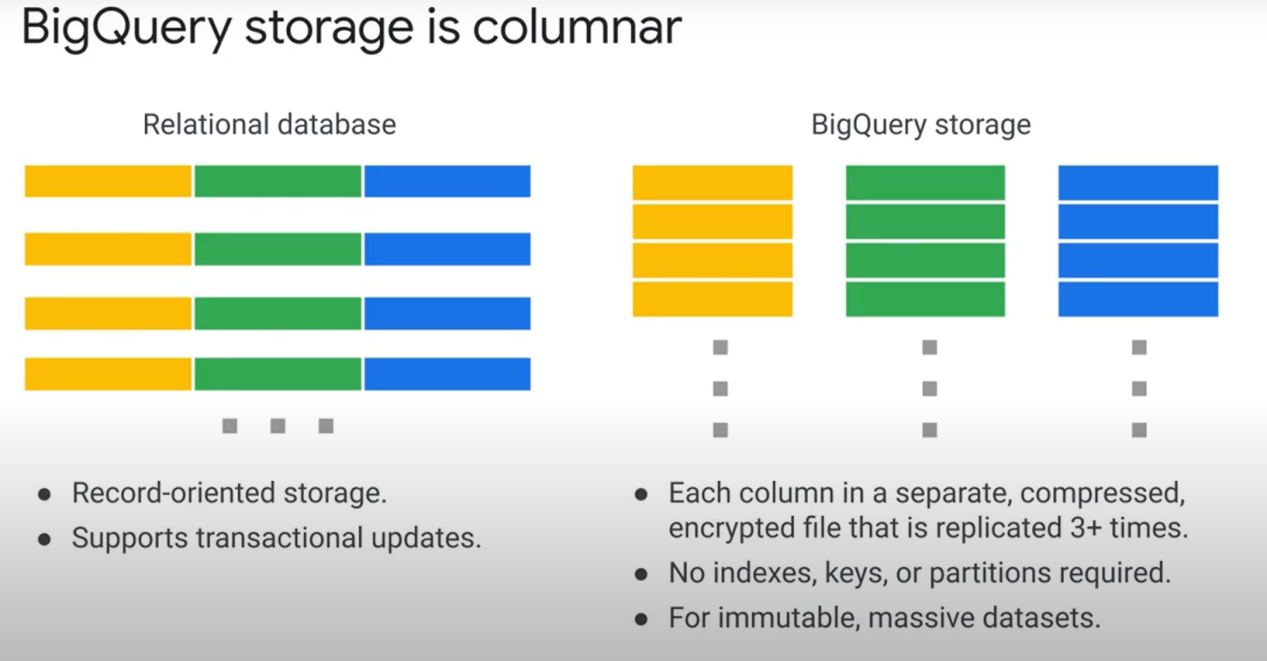
**- Cloud Spanner is ACID compliant**

- BQ mainly charges you for queries (and storage, but mainly for the queries). Spanner

charges you for storage and computing.

- BQ has more integration with data studio, BI engine, Apache, etc...

- Spanner has more integration with Cloud Storage, Cloud Run, Terraforms Liquibase, etc...



=> BQ is very good at reading and appending, but bad at updating/deleting. Good for immutable datasets.

Note for costs calculus:

1000 processing units = 1 node. For cloud spanner, you are charged by:

(number of nodes \* total hours of use), with number of nodes= num processing units/1000

Note 2: **«AVRO»** is a technology, mainly used in Hadoop, for **serialization/deserialization**

**BigQuery VS BigTable:**

\*BigTable: only 1 index per table. All data is stored in **ascending** order: if using timestamps as index, and if you need to often query for the most recent data, consider reversing the timestamps => New problem: rows with the same timestamp will be adjacent to each other, thus, **they will be in the same table**. That means the processing isn't distributed in the end, and the **node** containing that specific **tablet** will have to do most of the work...

\*BigTable is NoSql

**Storage element of each technology:**

**\* Data proc ( Spark )** use RDD (Resilient Distributed Datasets) to store objects. Rdds hide complexity and allow spark to make decisions on your behalf

\* **Classic Hadoop**: **File system (hadoop, HDFS)**, easy to find elements by name,

\* **Cloud SQL, Cloud Spanner, BigQuery: relational database** = easier to find data by logic (like sql queries)

\* **Bigquery**: Storing in a processing system = easier to process and transform the data

\* **BigTable: NoSQL** wide column

\* Cloud Datastore: NoSql document

\* Dataflow: PCollections. Note that dataflow uses the same code for both batch (bounded) and streaming (unbounded) data

Hadoop VS Dataflow: 1- many hadoop workloads can be run more easily and are easier to maintain with cloud dataflow but p collections and rdds are not identical so existing code has to be redesigned and adapted to run in the cloud dataflow

1. In dataflow, the creation of clusters is automatically taken care of for you, unlike in hadoop