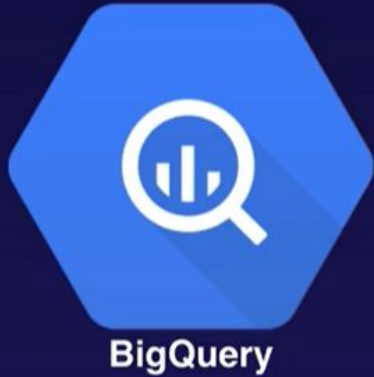


GCP

Data Analytics with BigQuery

BigQuery basics

Introducing BigQuery



- **Peta-byte scale, serverless, highly-scalable cloud enterprise data warehouse**
- **In-memory BI Engine (BigQuery BI Engine)**
- **Machine-learning capabilities (BigQuery ML)**
- **Support for geospatial data storage and processing**

key features

1

High Availability

Automatic data replication to multiple locations with high availability.

2

Supports Standard SQL

BigQuery supports a standard SQL dialect which is ANSI:2011 compliant.

3

Federated Data

BigQuery able to connect to, and process, data stored outside of BigQuery (external sources).

4

Automatic Backups

BigQuery automatically replicates data and keeps a seven-day history of changes.

5

Governance and Security

Fine-grained identity and access management.
Data encrypted at rest and in transit.

6

Separation of Storage and Compute

Leads to a number of benefits including ACID-compliant storage operations, cost-effective scalable storage and stateless resilient compute.

Interacting with BigQuery



BigQuery

- **Web console**
- **Command-line tool (bq)**
- **Client libraries**
 - C#, Go, Java, Node.JS, PHP, Python and Ruby

Managing Data

Project

Dataset 1



Native tables



External tables



Views

Dataset 2



Native tables

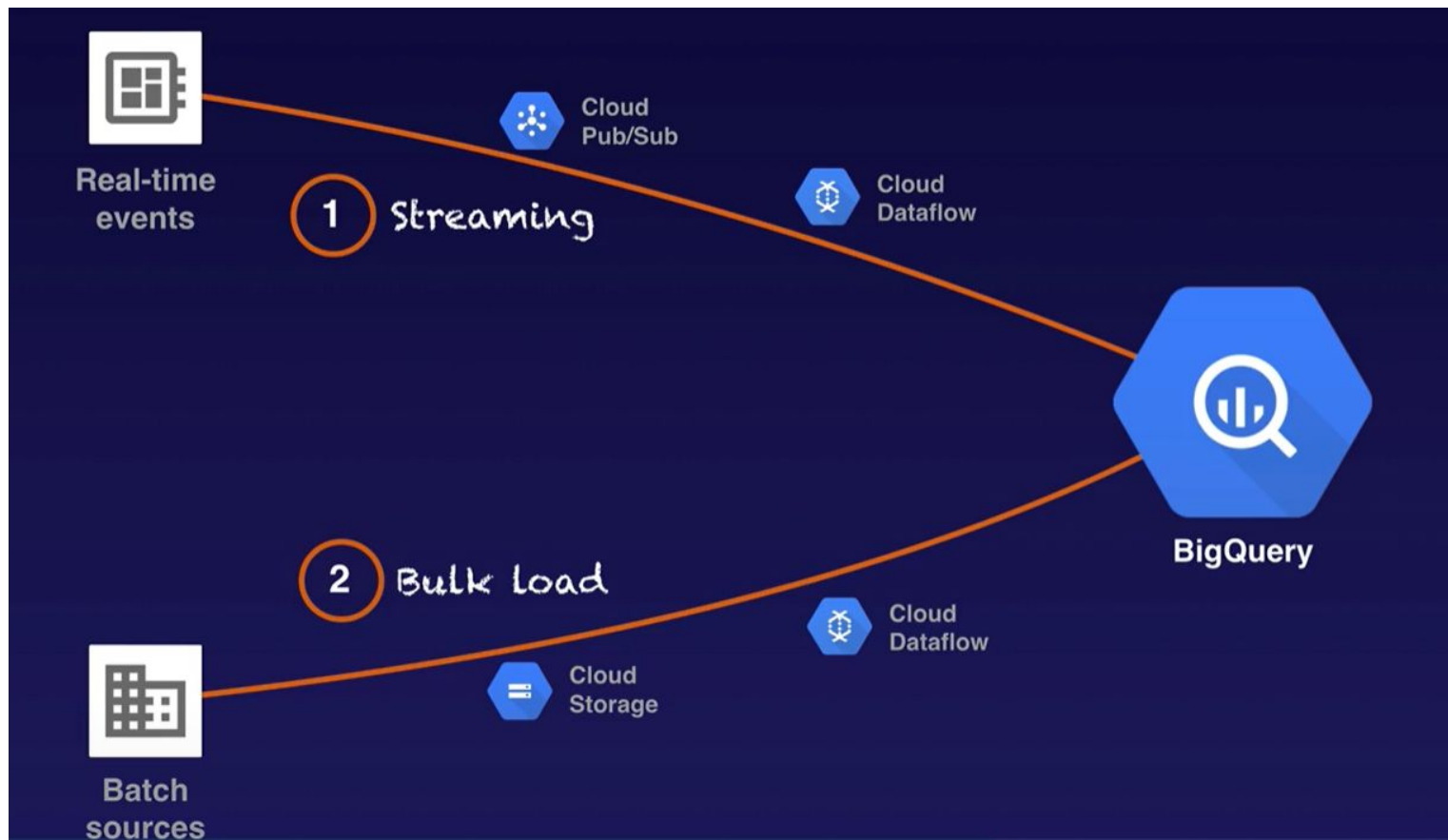


External tables



Views

Data ingestion



TheBigQuery SQL Dialect

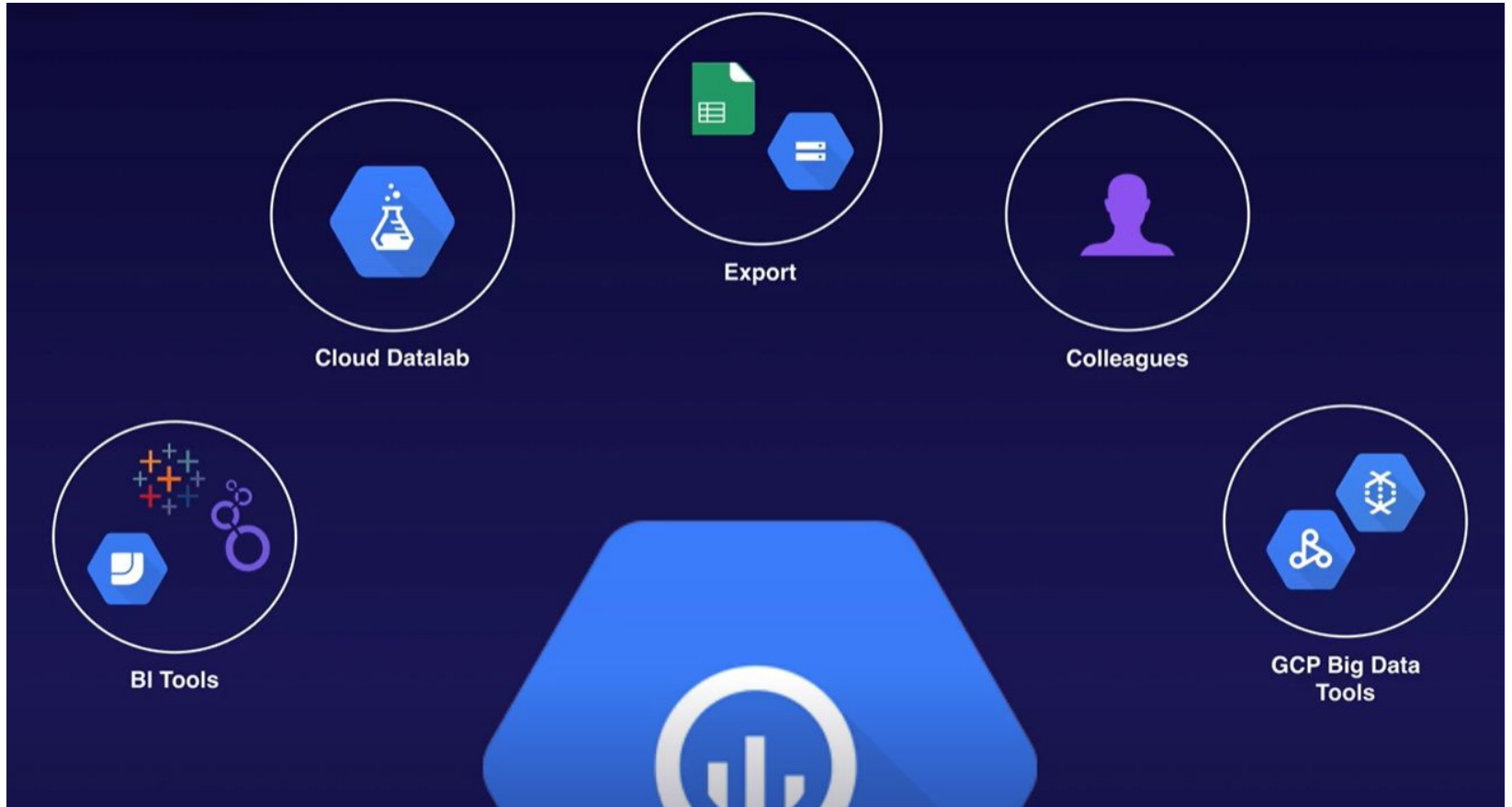
Legacy SQL

- Previously called BigQuery SQL
- Non-standard SQL dialect
- Migration to standard SQL is recommended

Standard SQL

- Preferred dialect
- Compliant with SQL 2011 standard
- Extensions for querying nested and repeated data

Using Data in BigQuery



BigQuery Jobs and Operations

Job: action that is run in BigQuery on your behalf (asynchronously)

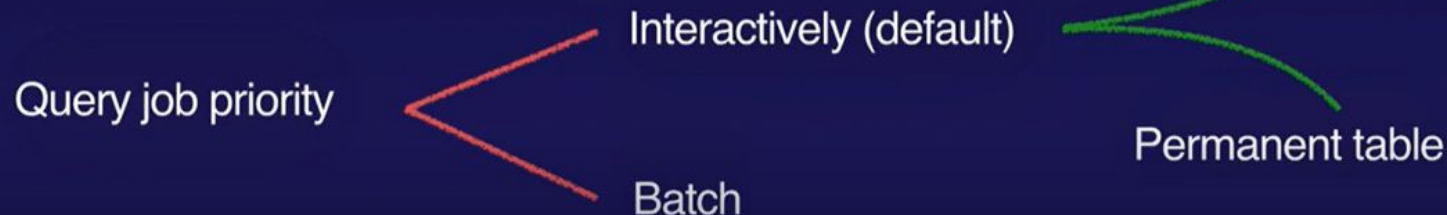


Table Storage in BigQuery

- **Capacitor** columnar data format
- Tables can be partitioned
- Individual records exist as rows
- Each record is composed of columns
- Table schemes specified at creation of table or at data load

	Column 1	Column 2	Column 3
Record 1			
Record 2			

Table A - Partition 1

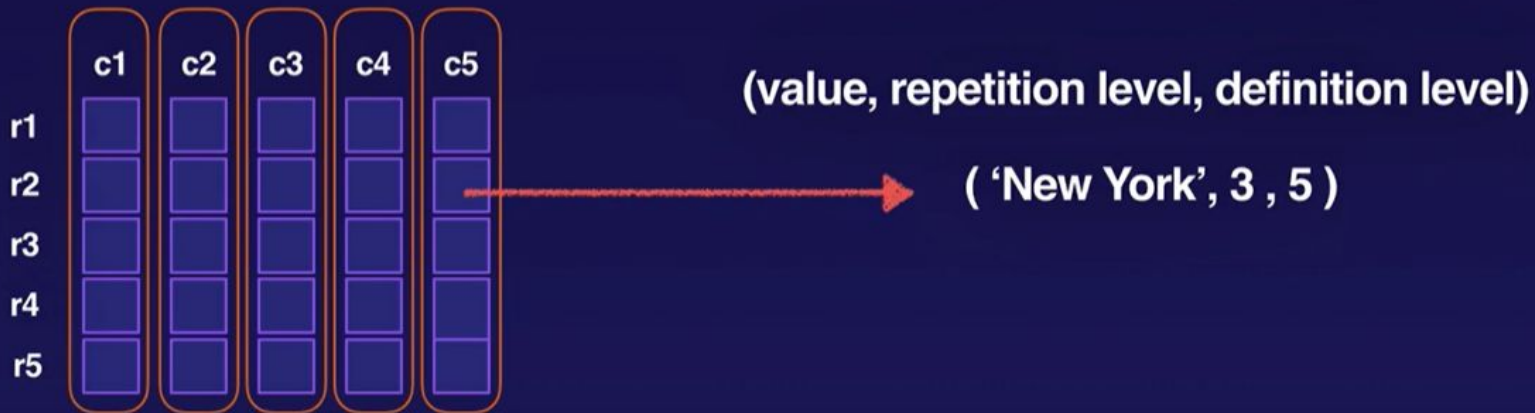
	Column 1	Column 2	Column 3
Record 3			
Record 4			

Table A - Partition 2

BigQuery Capacitor

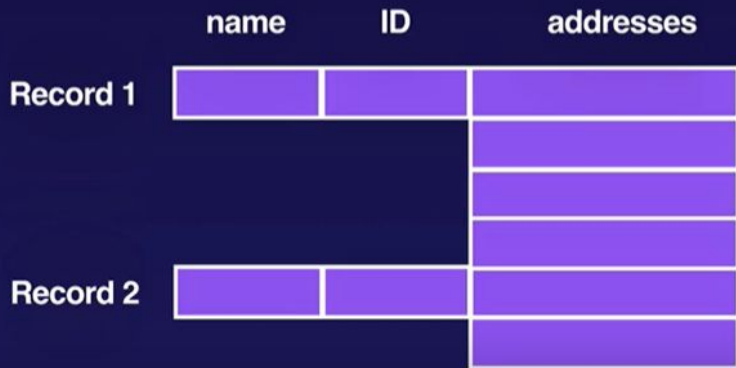
Capacitor storage system

- Proprietary columnar data storage that supports semi-structured data (nested and repeated fields)
- Each **value** stored together with a **repetition level** and a **definition level**



Denormalisation

- BigQuery performance optimised when data is denormalised appropriately
- Nested and repeated columns
- Maintain data relationships in an efficient manner
- **RECORD (STRUCT)** data type



Nested and repeated fields

- `id`
- `first_name`
- `last_name`
- `dob` (date of birth)
- `addresses` (a nested and repeated field)
 - `addresses.status` (current or previous)
 - `addresses.address`
 - `addresses.city`
 - `addresses.state`
 - `addresses.zip`
 - `addresses.numberOfWorkYears` (years at the address)

```
{  
  "id": "1",  
  "first_name": "John",  
  "last_name": "Doe",  
  "dob": "1968-01-22",  
  "addresses": [  
    {  
      "status": "current",  
      "address": "123 First Avenue",  
      "city": "Seattle",  
      "state": "WA",  
      "zip": "11111",  
      "numberOfWorkYears": "1"  
    },  
    {  
      "status": "previous",  
      "address": "456 Main Street",  
      "city": "Portland",  
      "state": "OR",  
      "zip": "22222",  
      "numberOfWorkYears": "5"  
    }  
  ]  
}
```

data Formats

Importing data

Supported formats



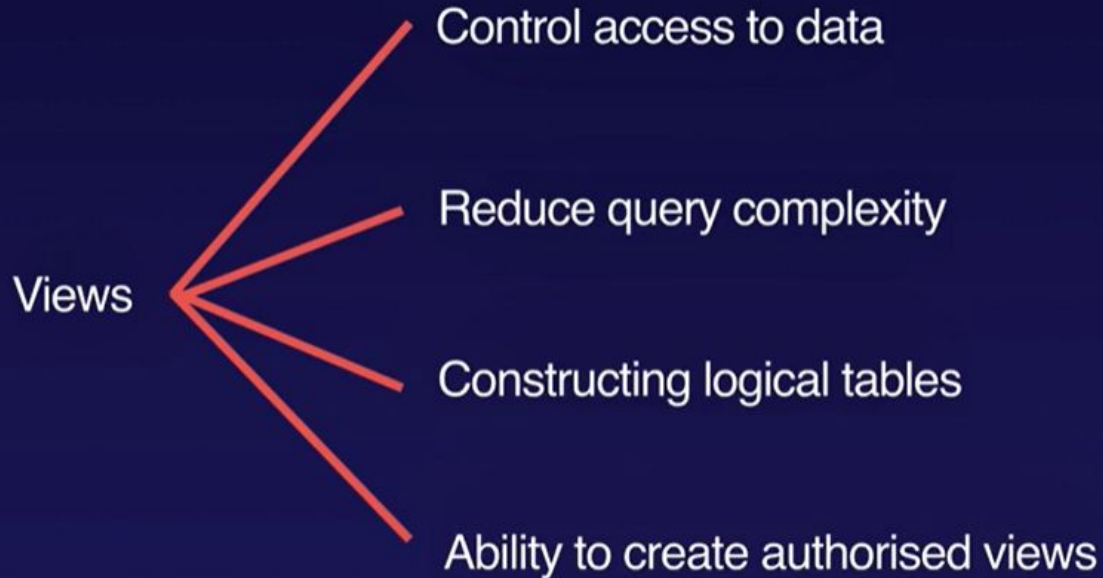
BigQuery Views

A view is a **virtual table** defined by a SQL query



BigQuery Views

Uses



BigQuery Views

Limitations

Views

Cannot export data from a view

Cannot use JSON API to retrieve data from a view

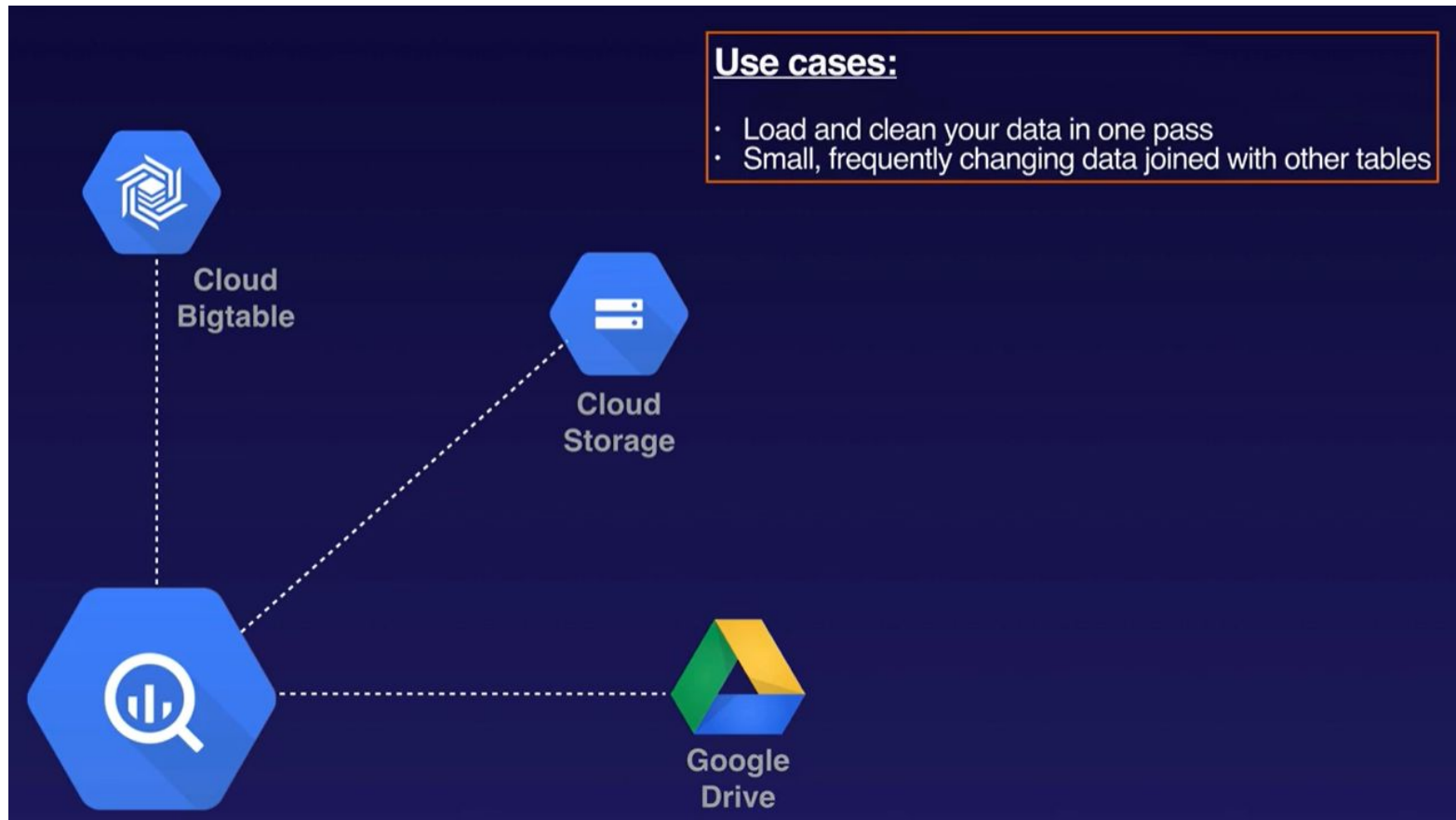
Cannot combine standard and legacy SQL

No user defined functions

No wildcard table references

Limited to 1,000 authorized views per dataset

External Data



External Data

Limitations

External data

No guarantee of consistency

Lower query performance

Cannot use TableDataList API method

Cannot run export jobs on external data

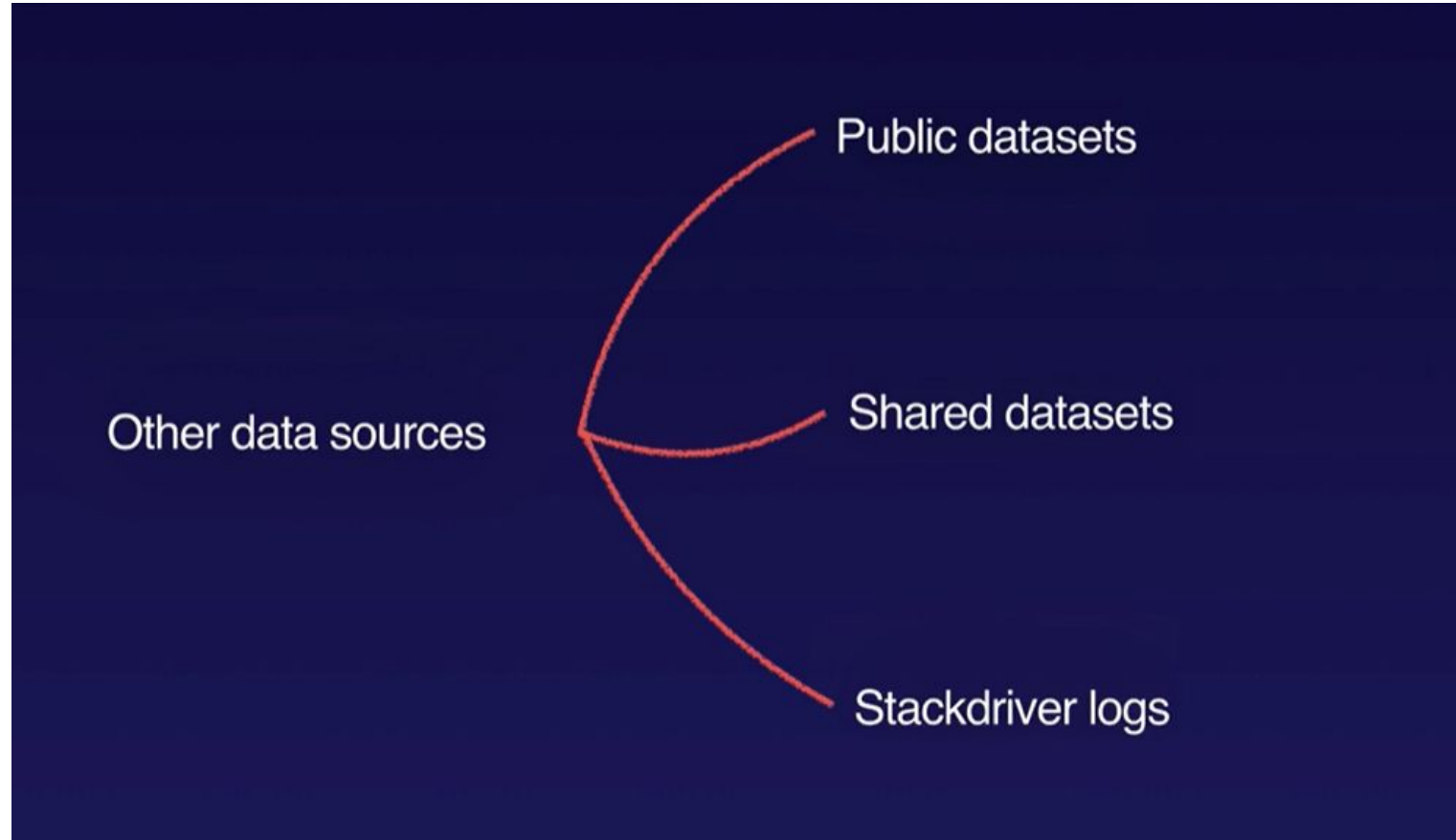
Cannot reference in wildcard table query

Cannot query Parquet or ORC formats

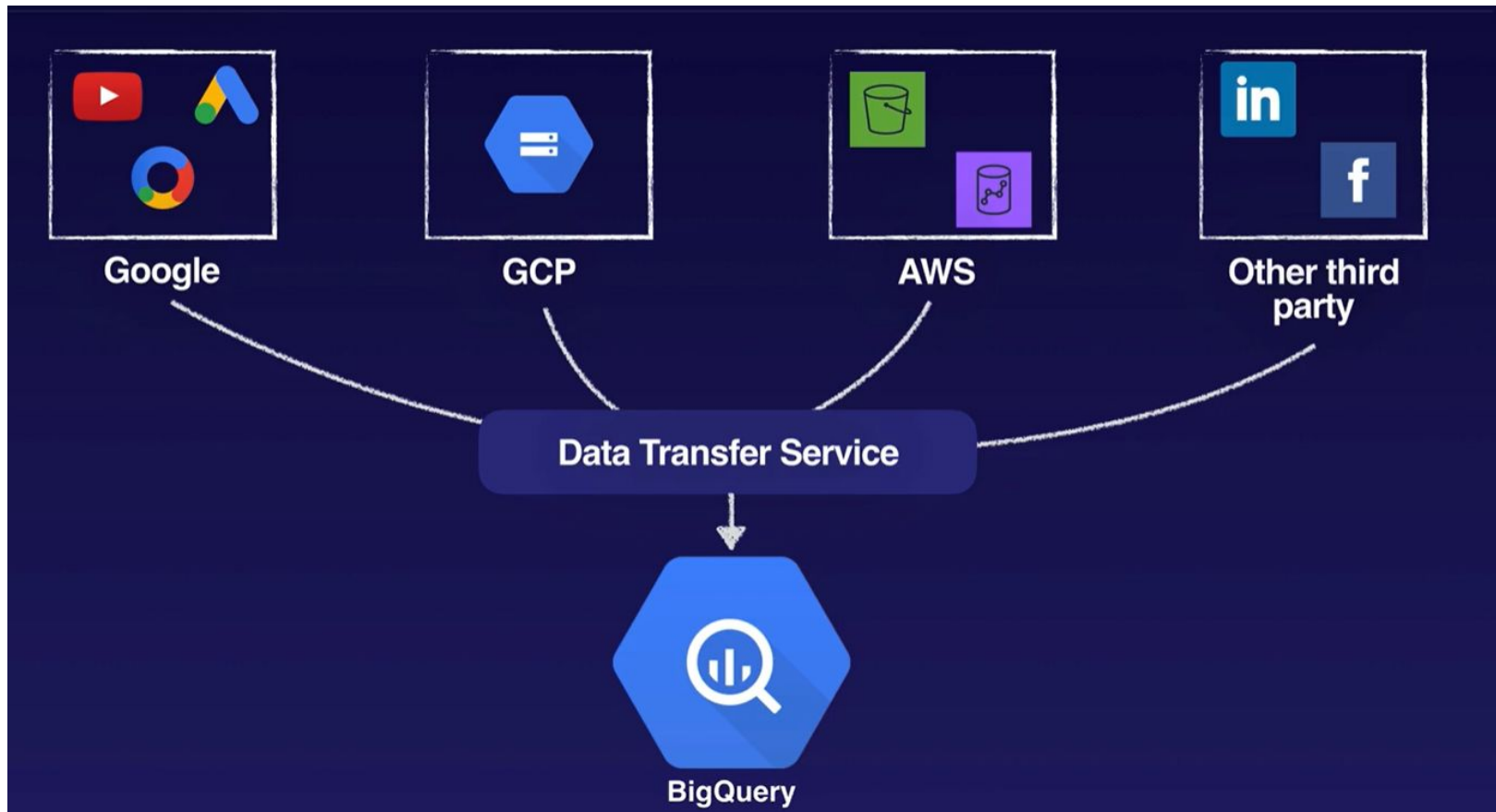
Query results not cached

Limited to 4 concurrent queries

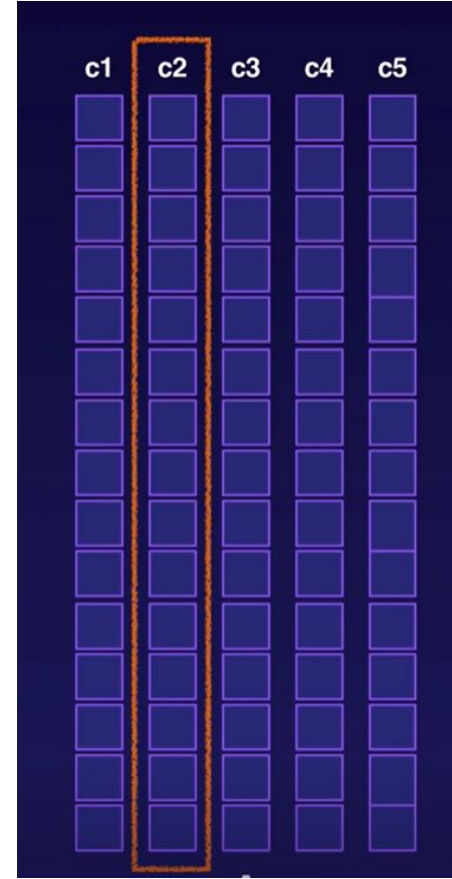
Other Data sources



Data Transfer service



Partitioning and Clustering

[illegible]

BigQuery Jobs and Operations



A - partition 1

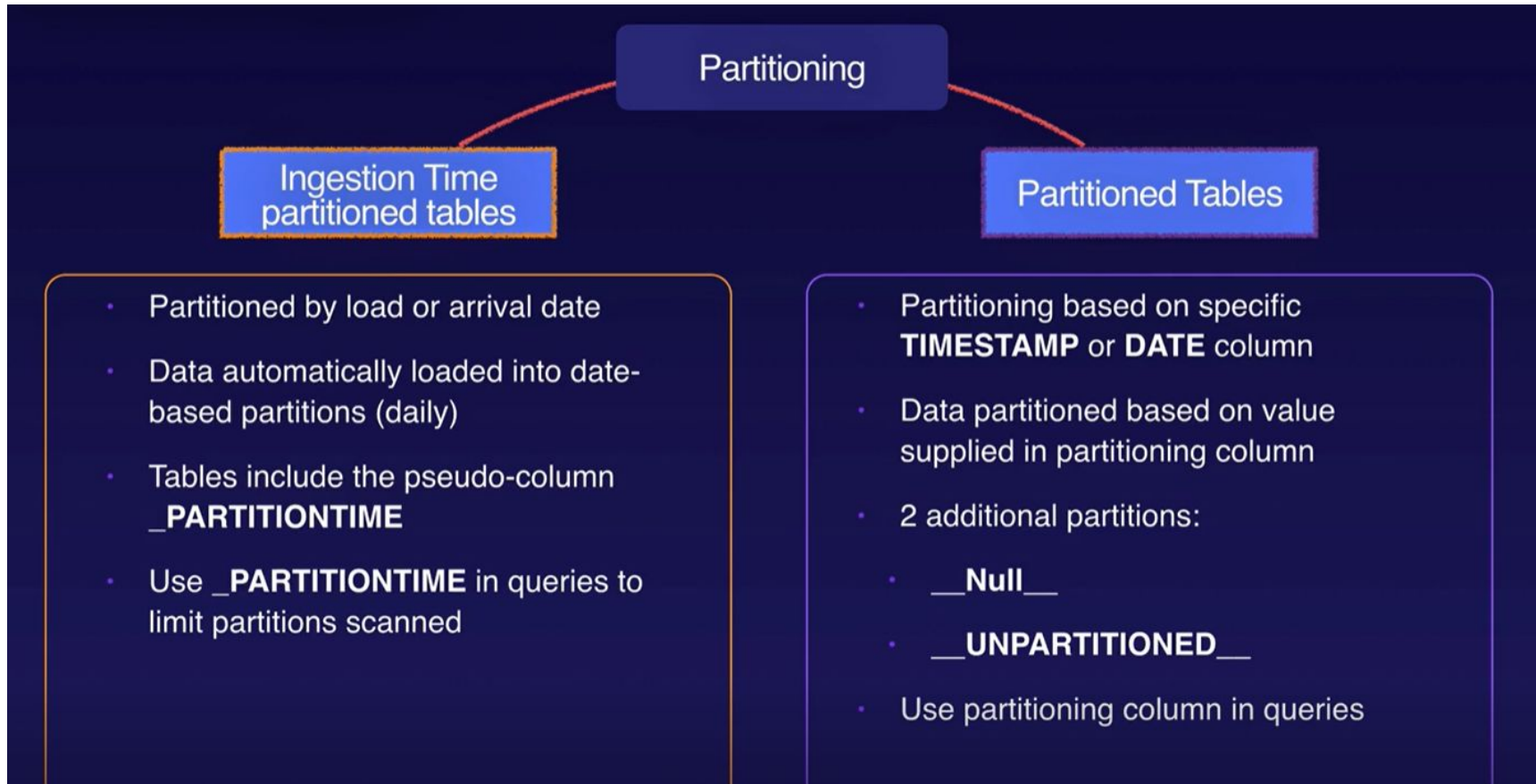


A - partition 2

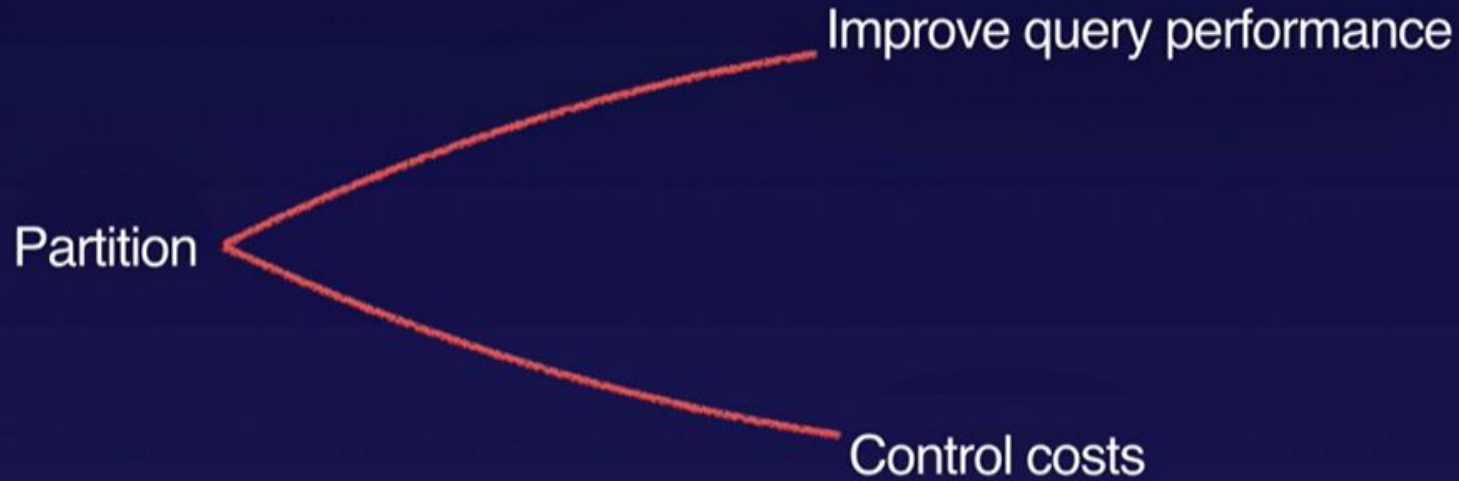


A - partition 3

BigQuery Jobs and Operations



BigQuery Jobs and Operations



BigQuery Jobs and Operations

A	B	C	D

Table_A
(2020-02-01)

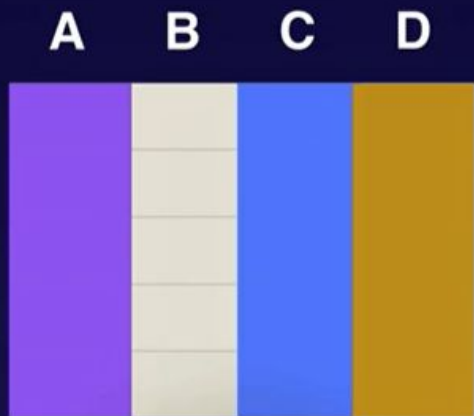
A	B	C	D

Table_A
(2020-02-02)

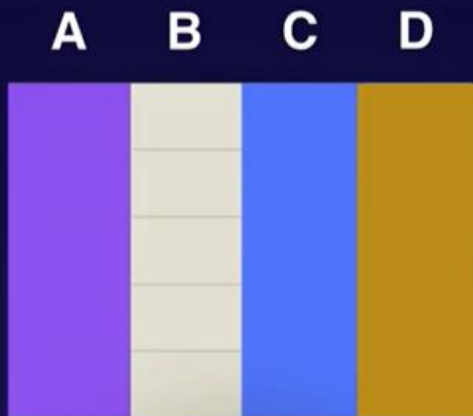
A	B	C	D

Table_A
(2020-02-03)

BigQuery Jobs and Operations



Table_A
(2020-02-01)

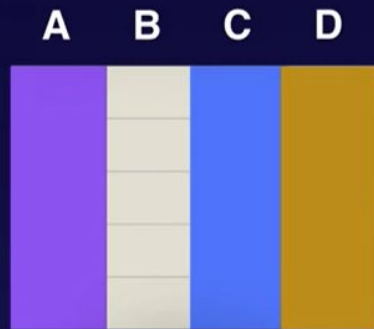


Table_A
(2020-02-02)

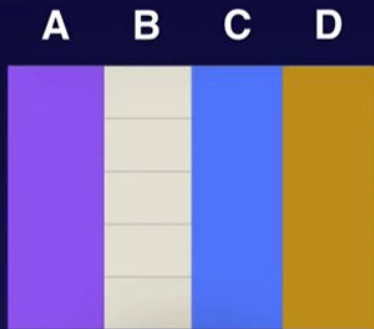


Table_A
(2020-02-03)

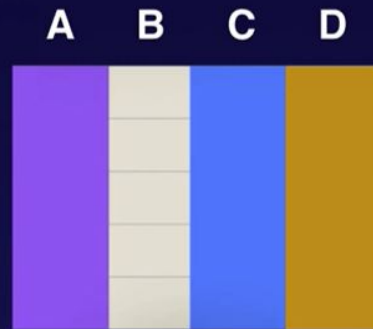
BigQuery Jobs and Operations



Table_A
(2020-02-01)



Table_A
(2020-02-02)



Table_A
(2020-02-03)


```
CREATE TABLE
`mydataset.ClusteredSalesData`
PARTITION BY
DATE(timestamp)
CLUSTER BY
customer_id,
product_id,
order_id
```



BigQuery Jobs and Operations

Limitations

Clustered tables

- 
- A diagram consisting of a central point from which seven red lines radiate outwards to the right, each pointing to a specific limitation of clustered tables.
- Only supported only for partitioned tables
 - Standard SQL only for querying clustered tables
 - Standard SQL only for writing query results to clustered tables
 - Specify clustering columns only when table is created
 - Clustering columns cannot be modified after table creation
 - Clustering columns must be top-level, non-repeated columns
 - You can specify one to four clustering columns

BigQuery Jobs and Operations

Querying Guidelines

- Filter clustered columns in the order they were specified

```
CREATE TABLE
  `mydataset.ClusteredSalesData`
PARTITION BY
  DATE(timestamp)
CLUSTER BY
  1 customer_id,
  2 product_id,
  3 order_id
```

```
SELECT
  SUM(totalSale)
FROM
  `mydataset.ClusteredSalesData`
WHERE
  customer_id = 10000
  AND product_id LIKE 'gcp_analytics%'
```



```
SELECT
  SUM(totalSale)
FROM
  `mydataset.ClusteredSalesData`
WHERE
  product_id LIKE 'gcp_analytics%'
  AND order_id = 20000
```



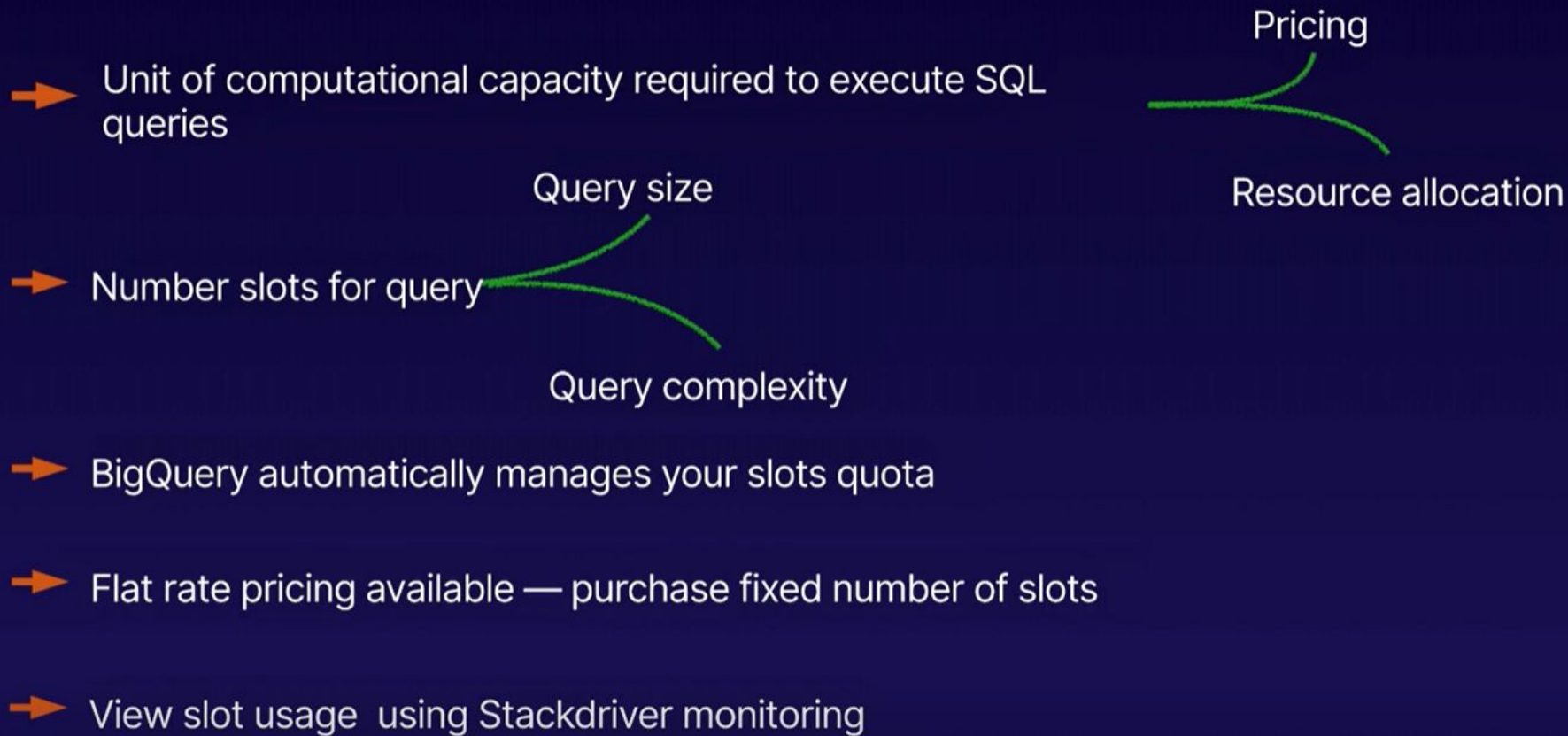
BigQuery Jobs and Operations

Querying Guidelines (continued)

- Avoid using clustered columns in complex filter expressions
- Avoid comparing cluster columns to other columns

BigQuery Best Practices

Slots



➔ Diagnostic query plan and execution timeline

➔ Declarative SQL statement

Query stages

Execution steps

➔ Query stage information

Stage overview

Step information

Stage timing classification

Timeline metadata

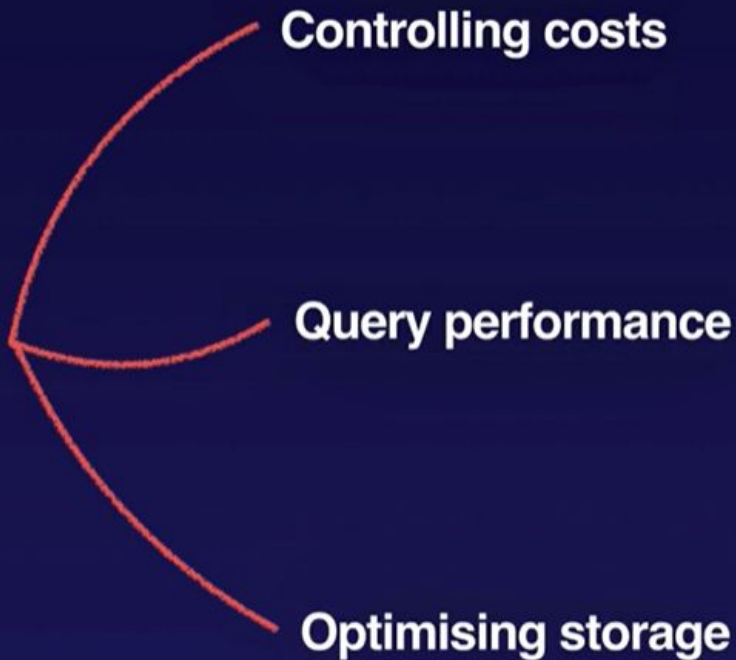
Best practices

BigQuery best practices

Controlling costs

Query performance

Optimising storage

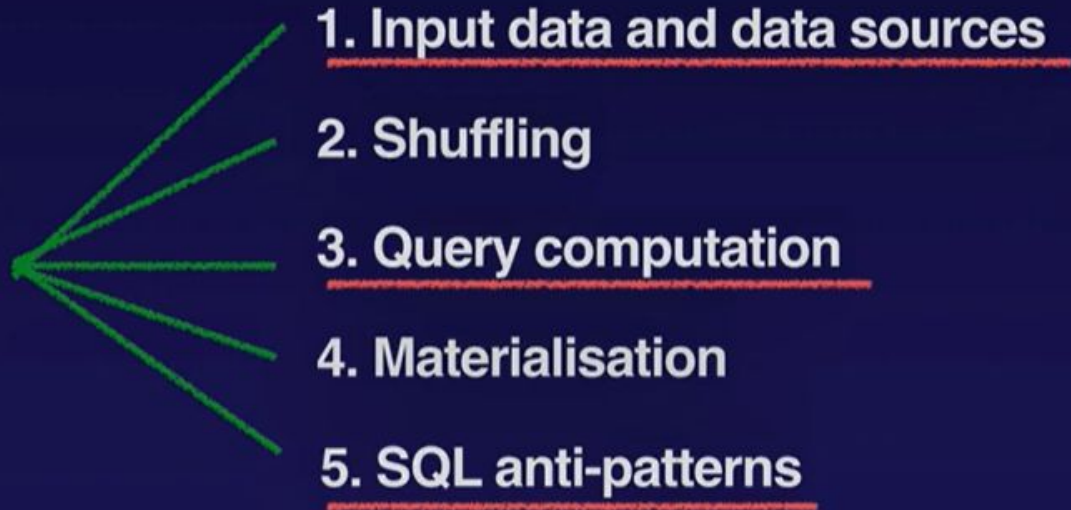


Cost Controls

- ➔ Avoid using **SELECT ***
- ➔ Use preview options to sample data
- ➔ Price queries before executing them
- ➔ Remember that using **LIMIT** does not affect costs
- ➔ View costs using a dashboard and query your audit logs
- ➔ Partition by date
- ➔ Materialise query results in stages
- ➔ Consider the cost of large result sets
- ➔ Use streaming inserts with caution

Query Performance Dimensions

Query performance

- 
- 1. Input data and data sources
 - 2. Shuffling
 - 3. Query computation
 - 4. Materialisation
 - 5. SQL anti-patterns

Query Performance: Input Data and Data Sources

- ➔ Prune partitioned queries
- ➔ Denormalise data whenever possible
- ➔ Use external data sources appropriately
- ➔ Avoid excessive wildcard tables

Query Performance: Query Computation

- ➔ Avoid repeatedly transforming data via SQL queries
- ➔ Avoid JavaScript user-defined functions
- ➔ Order query operations to maximise performance
- ➔ Optimise **JOIN** patterns

Query Performance: SQL Anti-patterns

- ➔ Avoid self-joins
- ➔ Avoid data skew
- ➔ Avoid unbalanced joins
- ➔ Avoid joins that generate more outputs than inputs (Cartesian product)
- ➔ Avoid DML statements that update or insert single rows

Optimising Storage


- Use expiration settings (tables automatically deleted at expiration)



Control Storage costs

Optimise use of storage space

- Take advantage of long-term storage



Lower monthly charges apply for data stored in tables or in partitions that have not been modified in the last 90 days

- Use the pricing calculator to estimate storage costs

Securing BigQuery

Roles

Primitive roles



Owner, Editor, Viewer

Predefined roles



Granular access

Service-specific

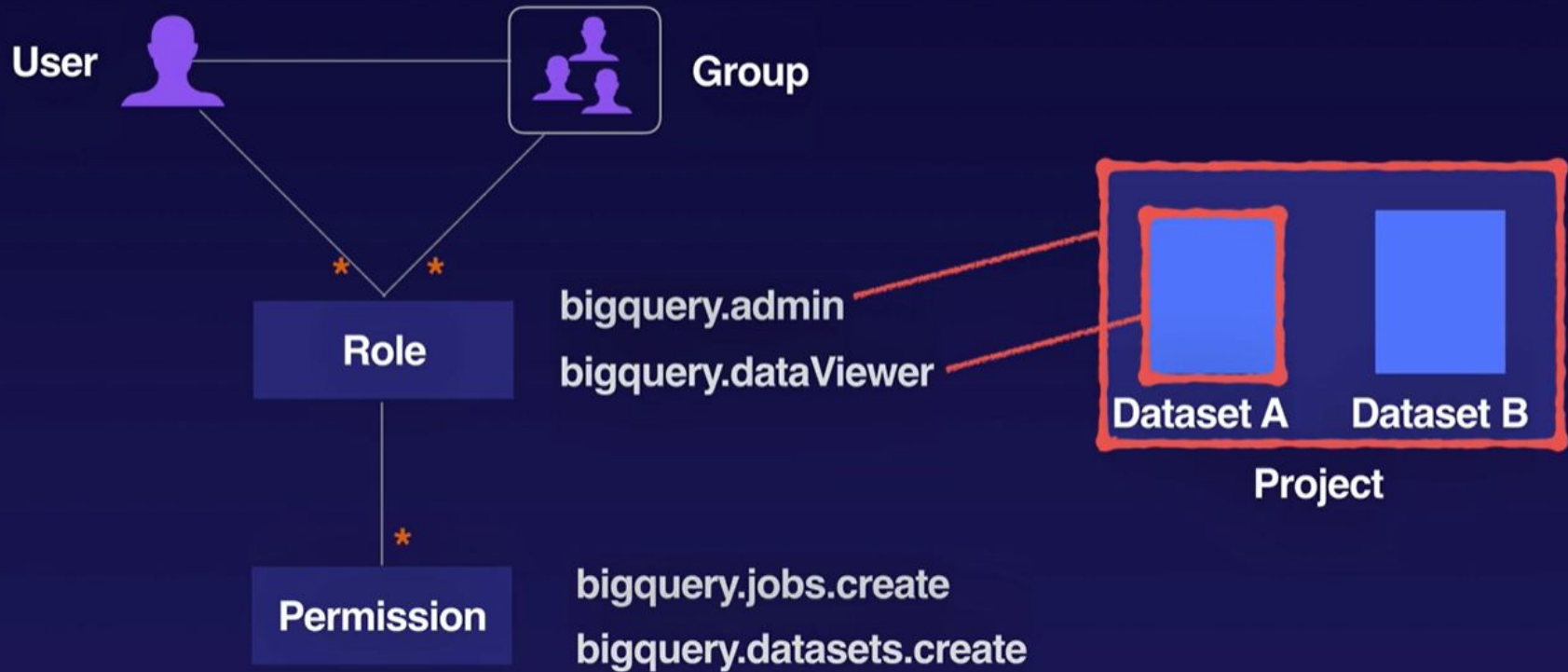
GCP managed

Custom roles



User managed

Roles



Handling Sensitive Data



Cloud Data Loss Prevention (Cloud DLP)

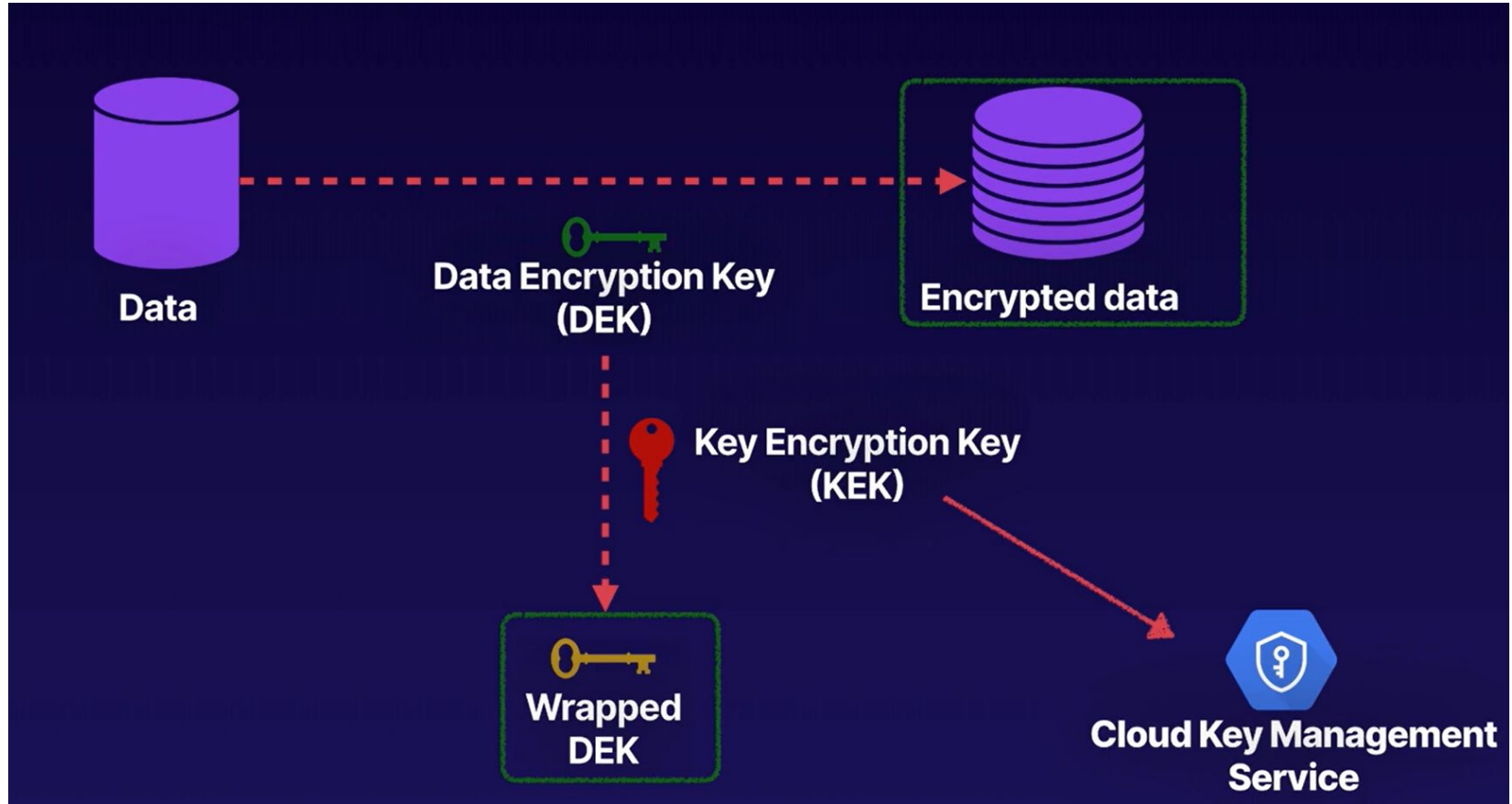
Cloud DLP



Data Loss Prevention API

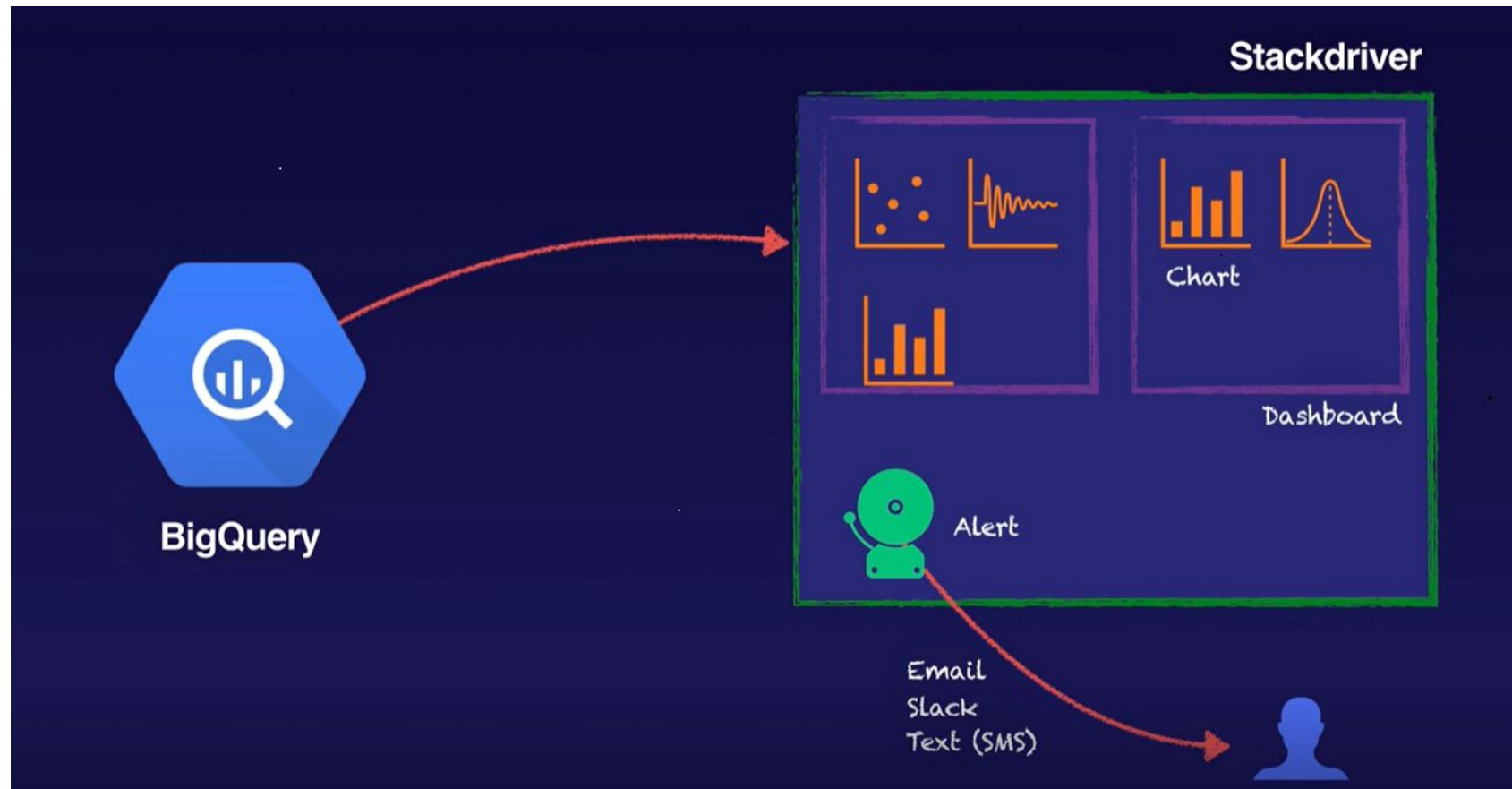
- **Fully managed service**
- **Identify and protect sensitive data at scale**
- **Over 100 predefined detectors to identify patterns, formats, and checksums**
- **De-identifies data using masking, tokenisation, pseudonymisation, date shifting and more**

Cloud Key Management Service

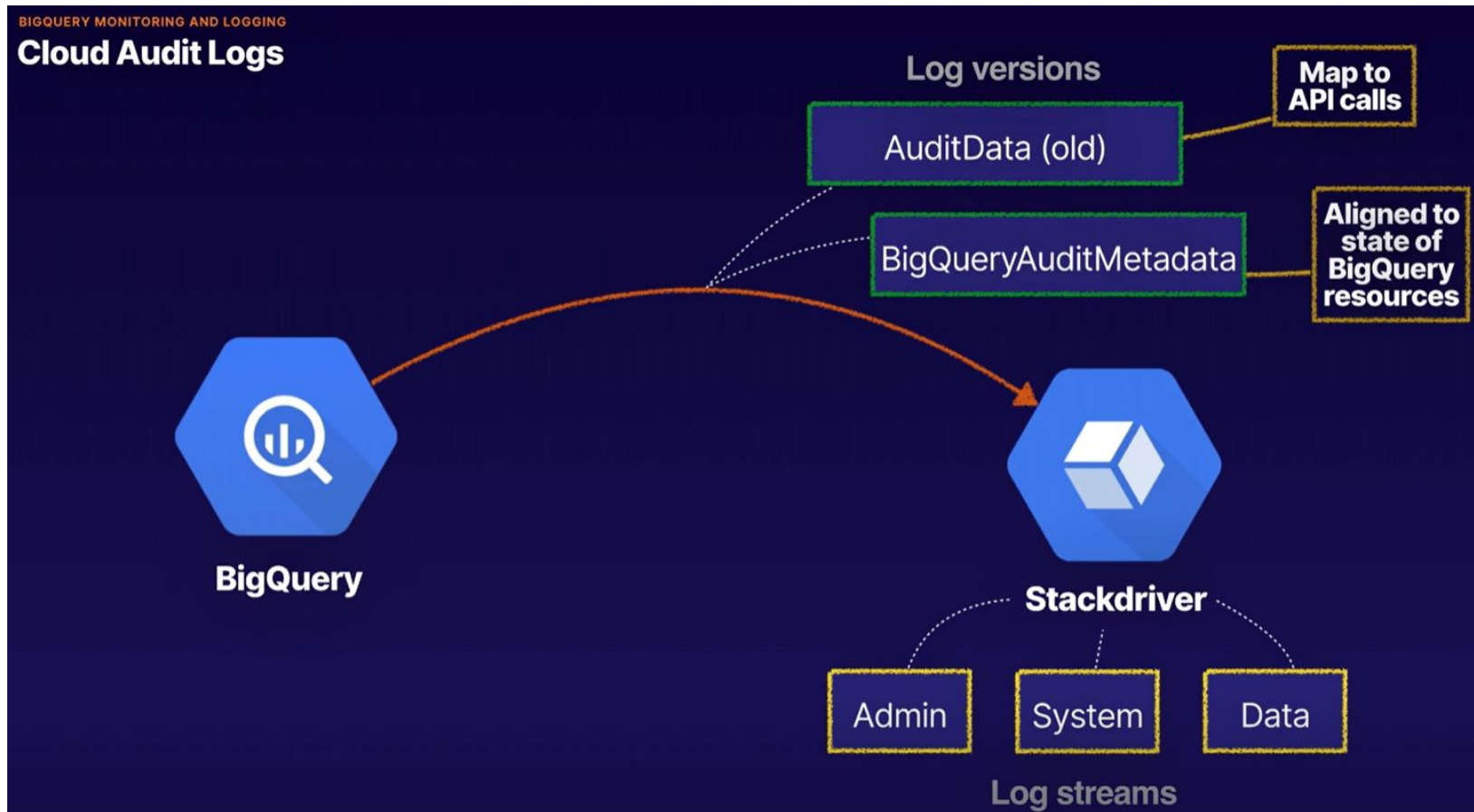


BigQuery Monitoring and Logging

BigQuery Monitoring



Cloud Audit Logs



Machine Learning with BigQuery ML

BigQuery ML



BigQuery ML

- **Web console (UI)**
- **bq command line tool**
- **BigQuery rest API**
- **Jupyter notebooks (Cloud Datalab) and other external BI tools**

ML Models

Linear regression



Binary logistic regression



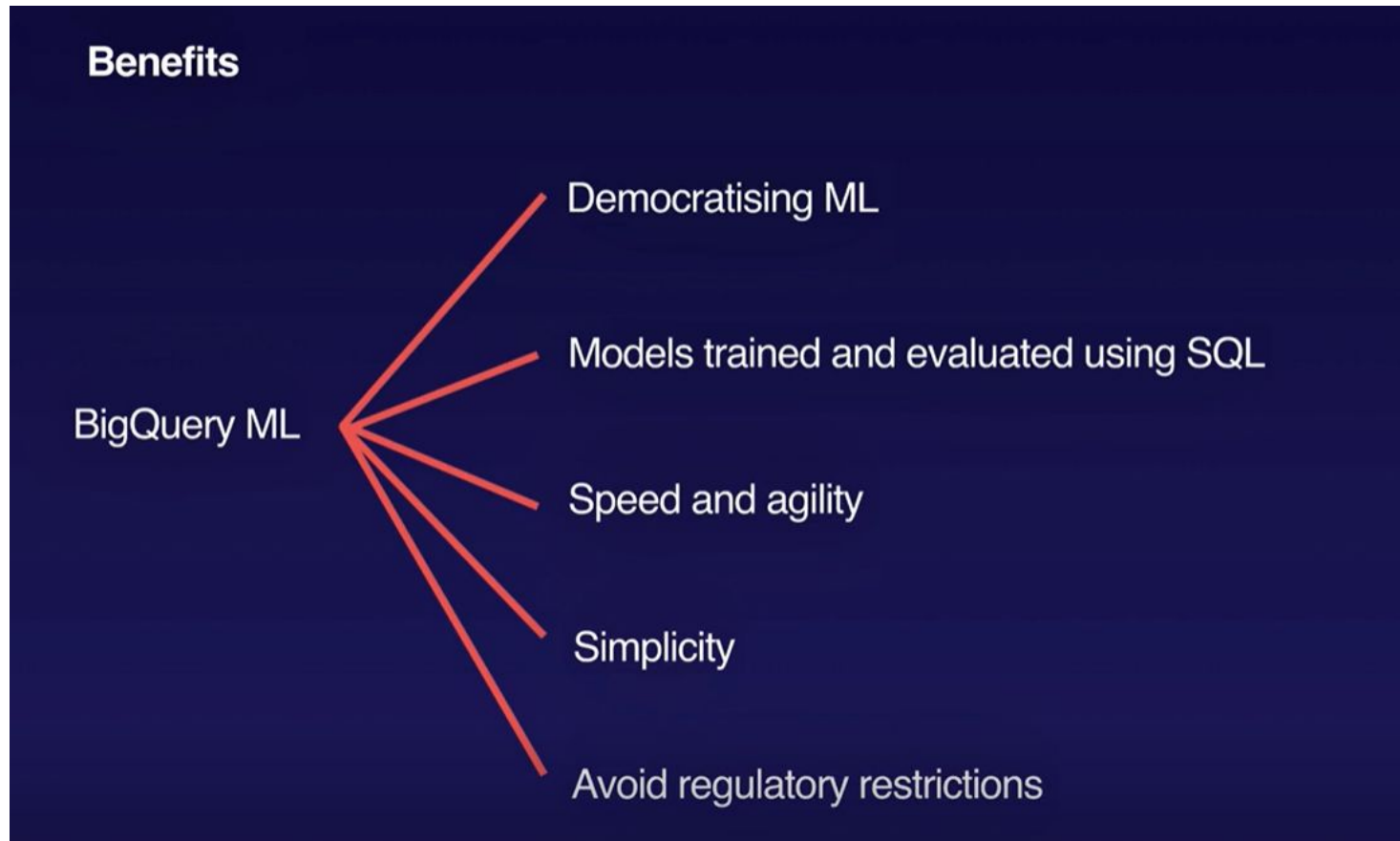
Multi-class logistic regression



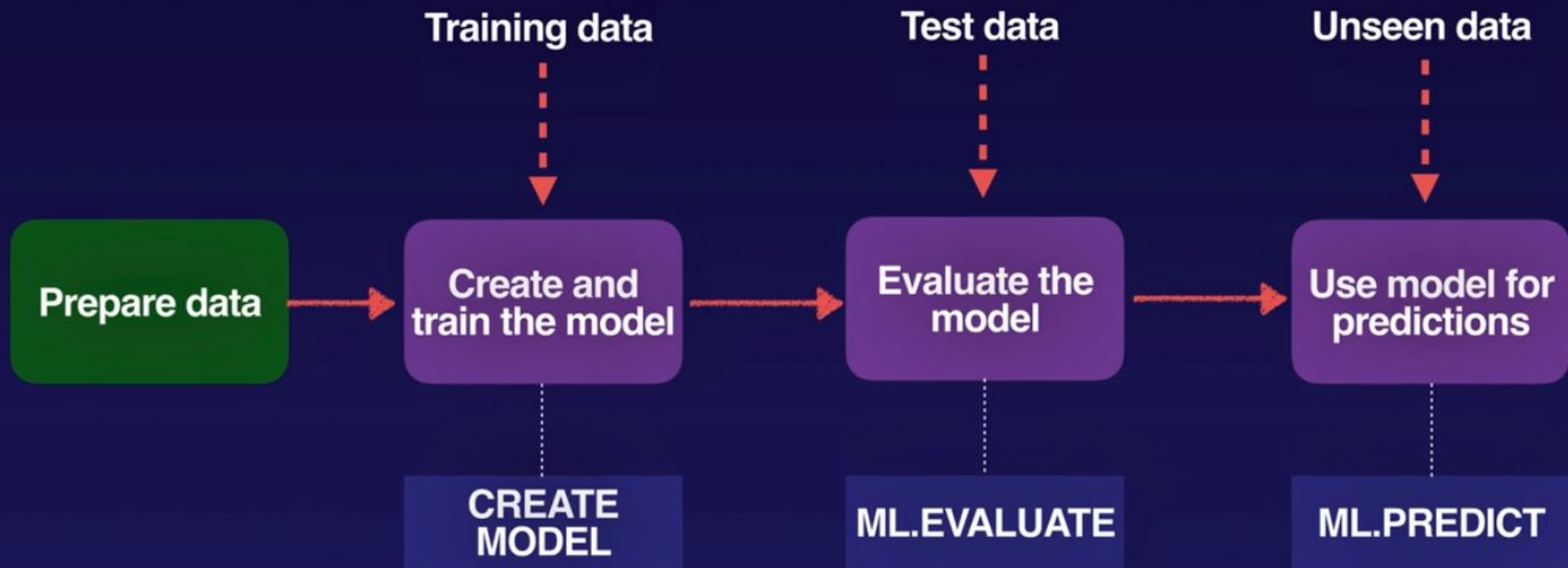
K-means clustering



BigQuery ML



BigQuery ML process



Thank you