Special Crafted by

Johanes Alexander

Training series no.
AAA.BBB.CC.DDD.EE

Updated
Q1.2018

Google BigQuery 101

Introduction into Google BigQuery: Conceptual and Practice

Trainer Bio

Johanes Alexander

Johanes Alexander is Lead Solutions Architect in GO-JEK's Business Intelligence and a final-year MBA student at Bandung Institute of Technology (ITB). His work is to design and build a reliable, scalable and accessible data infrastructure: to serve all people throughout the company towards data-driven decisions. Johanes is building the team and data infrastructure (pipelining, warehouse, visualization) from the scratch at the earlier days of the company. Currently almost thousand of people from business, product and engineering rely on the those system, in self service manner, to build the next big thing.

Email : alexander.v21@gmail.com
LinkedIn : https://www.linkedin.com/in/johanesa/

Agenda

- Conceptual:
  - Revisit: Data Warehouse
    - Concept: Data Lake? Data Warehouse?
    - Design: Snowflake or Star schema? Row-store or Columnar? Single node or distributed?
    - Operational: performance, scalability and reliability
  - Introduction to BigQuery
    - BigQuery at a glance — introduction, architecture, data type and how to use
- Practice:
  - Google BigQuery data loading using Python

Conceptual Revisit: Data Warehouse

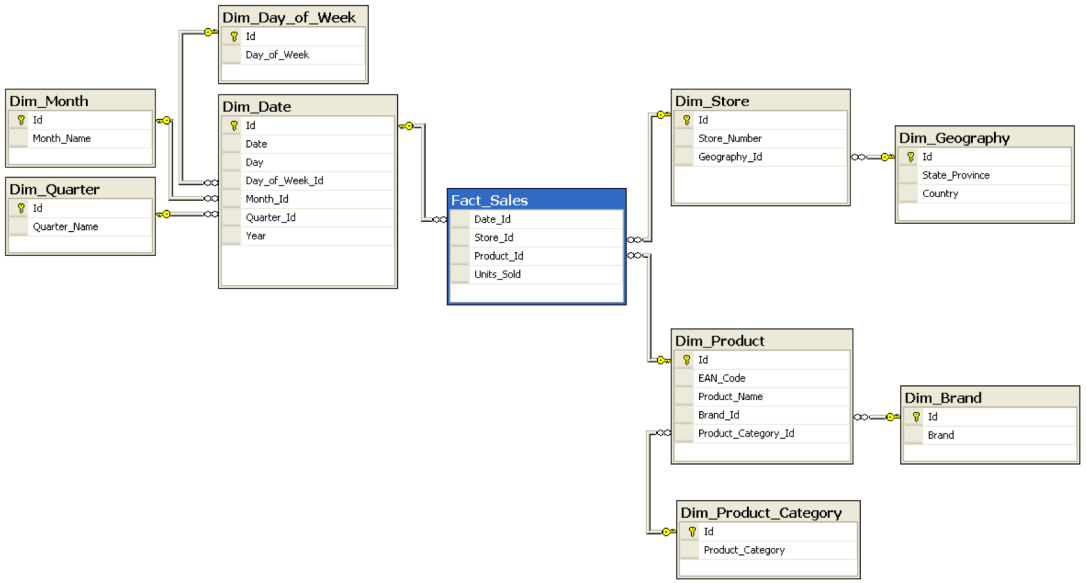
Concept: Data Store

*A **data lake** is a method of storing data within a system or repository, in its natural format, that facilitates the collocation of data in various schema and structural forms, usually object blobs or files — Wikipedia*

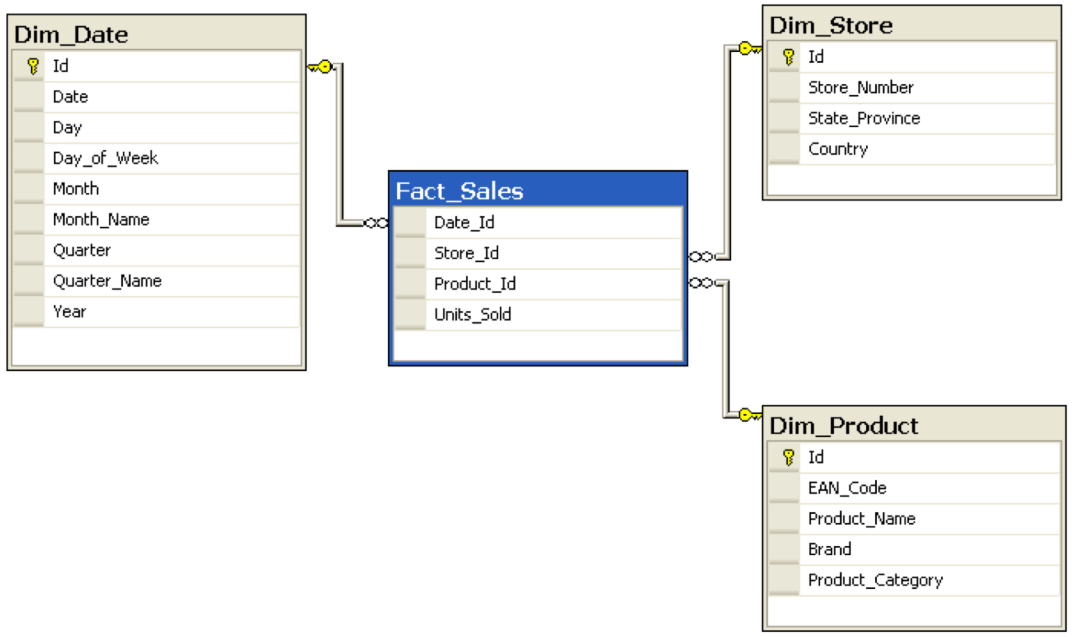
*In computing, a **data warehouse**, also known as an enterprise data warehouse, is a system used for reporting and data analysis, and is considered a core component of business intelligence — Wikipedia*

Design: Data Model

Snowflake Schema



Star Schema



Design: Database Type

Row-store

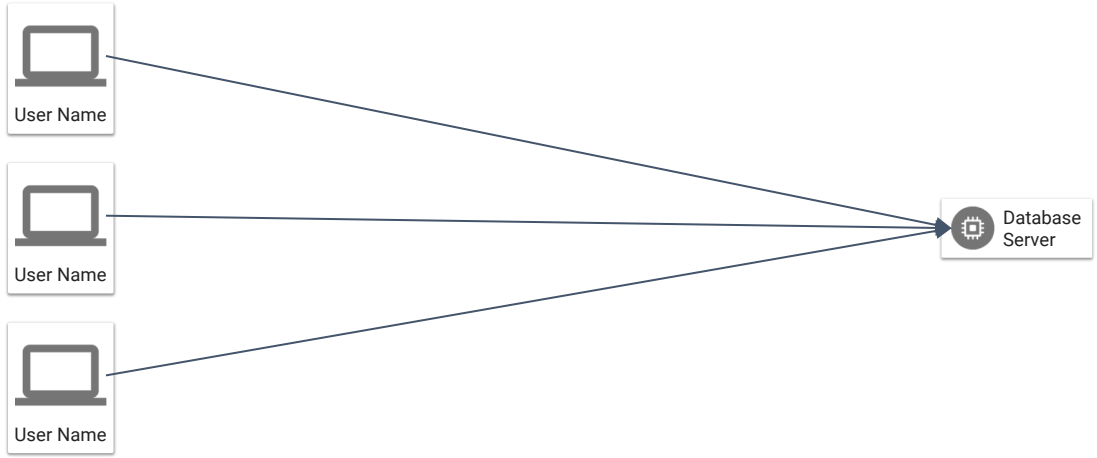
Column 1	Column 2	Column 3	Column 4	Column 5

Columnar

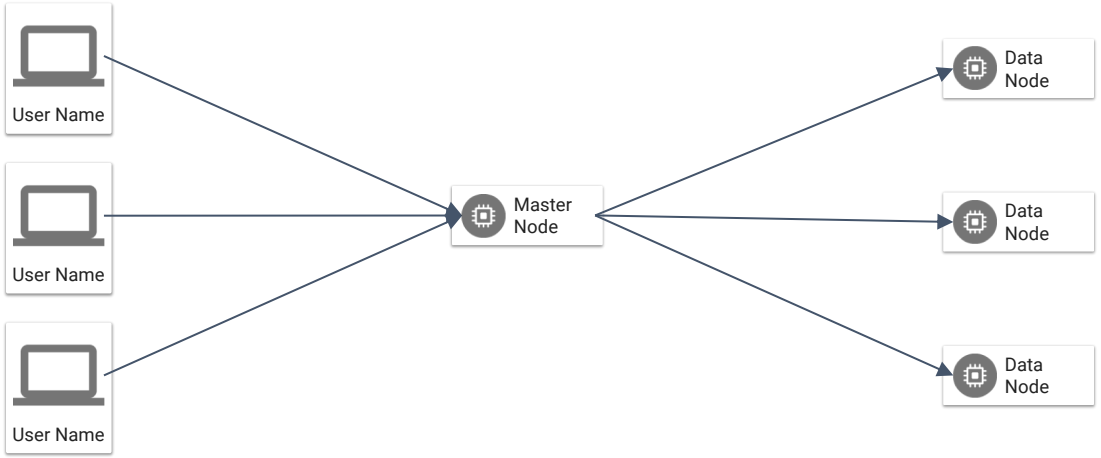
Column 1	Column 2	Column 3	Column 4	Column 5

Design: Infrastructure

Single-node Database



Distributed Database



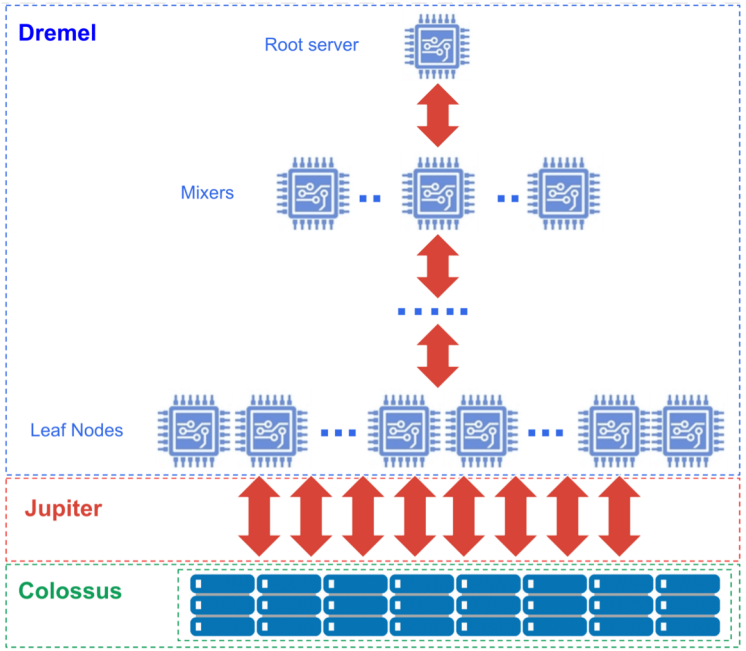
Operational

- Performance
  - *“The database is so slow! We need to scale!”*
- Scalability
  - *“Now we have a lot of system! We need to have high availability and a robust system!”*
- Reliability
  - *“Now the system is pretty robust! But we need more manpower to handle the operational thingy!”*

Introduction to BigQuery

At a Glance

- Fully Managed SQL Data Warehouse
- OLAP Analytics Engine
- Scale from GB to PB with zero operations
- Process terabytes of data in tens of seconds



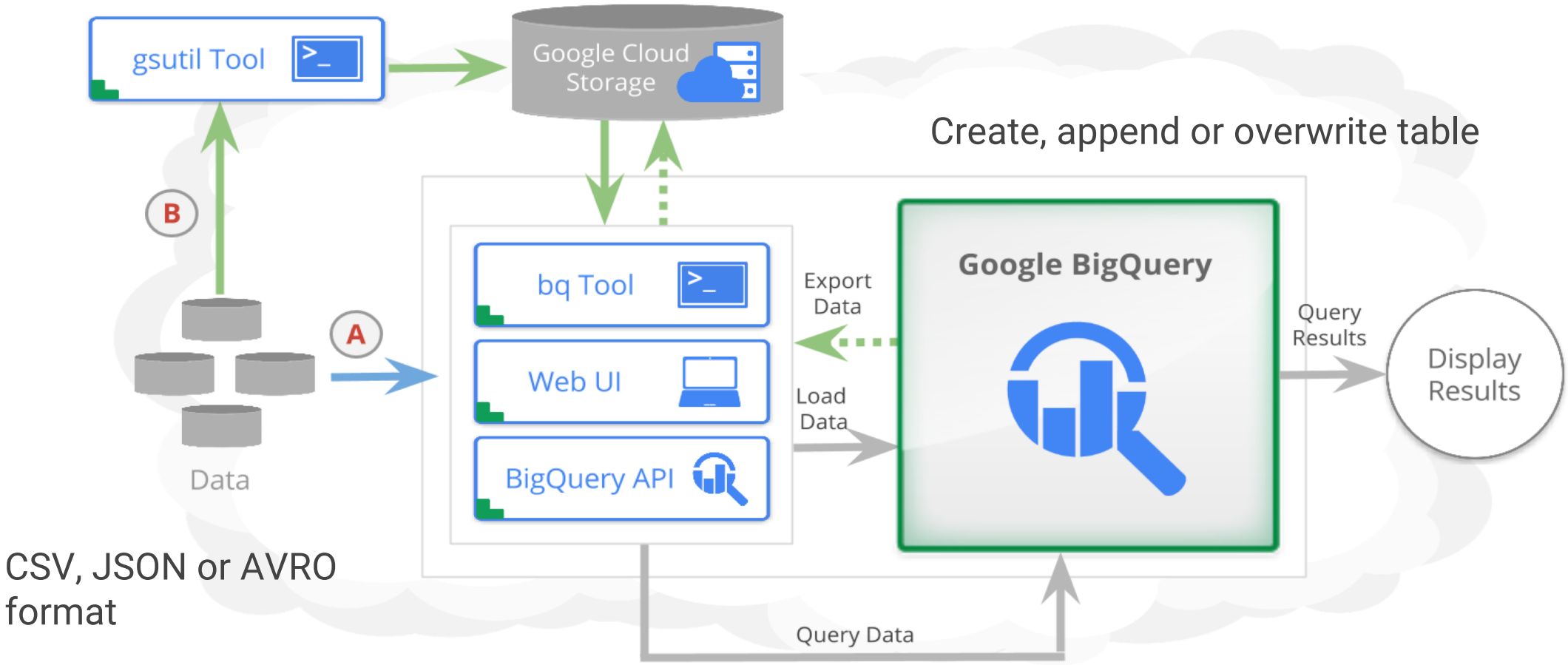
At a Glance — cont'd

	OLTP	OLAP
	Online Transaction Processing	Online Analytical Processing
Data Source	Operational	Historical
Focus	Updating/Retrieve	Reporting
Queries	Simple	Complex
Query Latency	Low	High
Google Cloud Platform Products	 Cloud SQL  Cloud Datastore  Cloud Spanner  BigTable	 BigQuery

At a Glance — cont'd

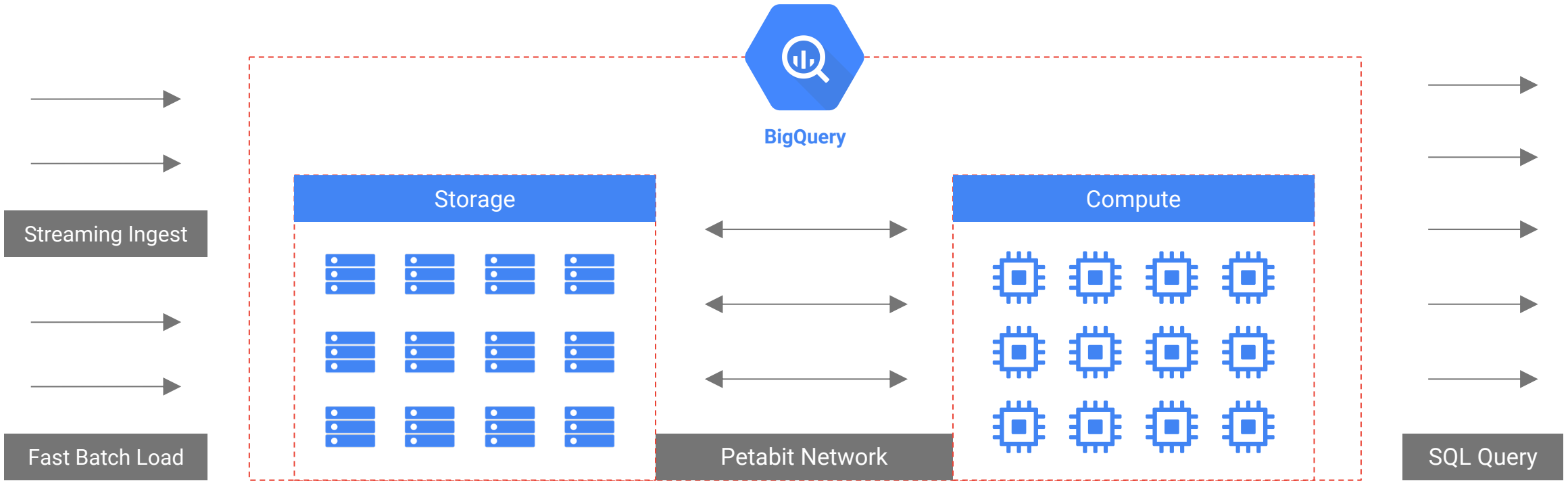
	Product	Interface	Query Latency	Typical Size	Storage Structure
	Cloud SQL	SQL	Low (ms)	< 10TB	Relational
	Datastore	Proprietary / NoSQL	Medium (10s of ms)	< 200TB	Document
	Bigtable	HBase API	Low (ms)	Terabytes - Petabytes	Key/Value
	BigQuery	REST / SQL / WebUI	High (s)	Terabytes - Petabytes	Columnar
	Spanner	SQL	Low (ms)	Terabytes	Relational

At a Glance — cont'd



CSV, JSON or AVRO format

At a Glance — cont'd



At a Glance — Goal

At a Glance — Data Type

- Support simple/primitive data type (integer, string, etc)
- Support complex data type (array, struct)
- JSON structure compatible

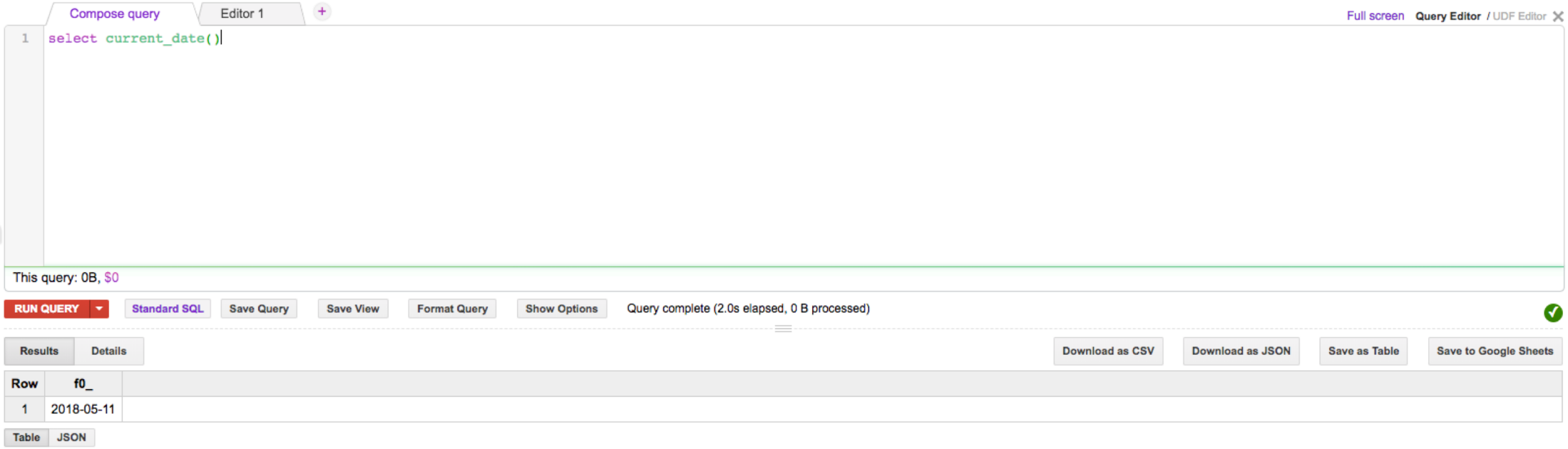
```
{
  "nama": "Budi",
  "umur": 22,
  "nomor_telepon": [
    6281234567890,
    6282233445566
  ],
  "keluarga": {
    "ayah": "dede",
    "ibu": "siti"
  }
}
```

keluarga	RECORD	NULLABLE	Describe this field...
keluarga.ibu	STRING	NULLABLE	Describe this field...
keluarga.ayah	STRING	NULLABLE	Describe this field...
nomor_telepon	INTEGER	REPEATED	Describe this field...
umur	INTEGER	NULLABLE	Describe this field...
nama	STRING	NULLABLE	Describe this field...

Row	keluarga.ibu	keluarga.ayah	nomor_telepon	umur	nama
1	siti	dede	6281234567890	22	Budi
			6282233445566		

At a Glance — How to Use

Just write the query!



BigQuery Data Loading with Python

Load the Data

Batch

- Supported format:
  - CSV, JSON, Avro, Parquet, Google Sheets
- Source:
  - File upload, Google Drive, Google Cloud Storage
- Via gcloud SDK, BigQuery Client API

Stream

- Via BigQuery Client API
- Event timestamp only from *30 days before and 5 days after*

Batch

Based on best practice, below is the steps to load the data:

- *Data is transformed to JSON* → clear data structure and type
- *Data is stored in Google Cloud Storage* → easy maintenance (no-ops), performance consideration during high volume loading
- *Can be loaded easily either through* `bq` *command line or using API* → freedom of choice
- *Data inserted into table with partition* → performance and cost consideration

Streaming

Based on best practice, below is the steps to load the data:

- *Data is transformed to JSON* → clear data structure and type
- *Data is passed through Pub/Sub (or another messaging bus)* → *for replayability consideration*
- *Data inserted into table with partition* → performance and cost consideration

Thank You