Week-2 Data Warehouse

Consolidate — Clean, accurate.

query = fast

- GB to PB
- Serverless & no ops
- ecosys. visualization & reporting tools
  - " of ETL & data processing tools
- Up to minute data
- ML
- Security & collab.

- GB to PB — Big query
- Serverless & no ops
- ecosys. visualization & reporting tools
  - " of ETL & data processing tools
- Up to minute data
- ML
- Security & collab.

- X Data aging
- X Storage management

- fault recovery
- Query engine optimization
- Hardware
- Updates

Big query = table-oriented (OLAP)

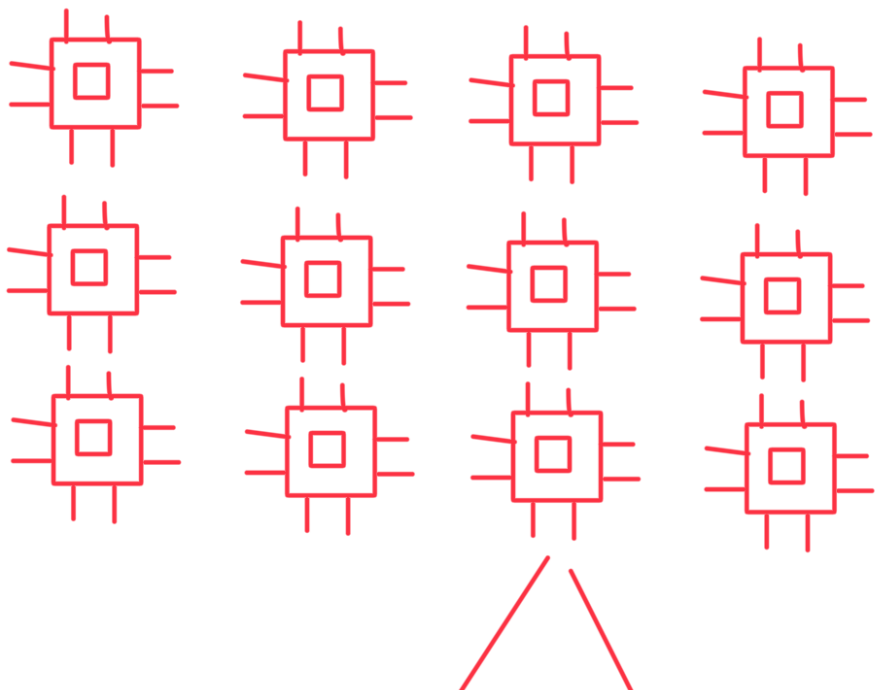
Storage engine Analytic engine

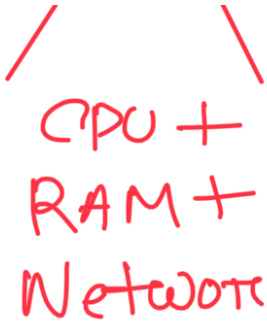
Run length encoding

Dictionary encoding

2000 slots

μ-service architecture





Bigquery Services

Multiple dataset

billing ⟶ project

Query ↘ IAM

Region ⟶ multi zone

Google managed encryption key

predefined roles

row level security ⇕ policy

authorised view

Analytics UI

view → virtual table

materialized view

SQL

pricing calculator

EL

ELT
ETL

CSV
AVRO
Newline-delimited_JSON
DATASTORE_BACKUP
PARQUET
ORC

CSV
JSON
Avro
Parquet
ORC

Dataflow

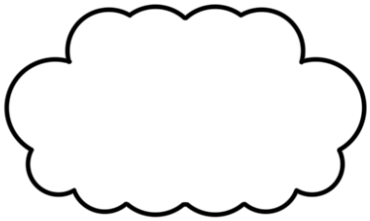
Saas
Bigquery Data Transfer

#Automate execution

point-in-time → 7 days

Transferring data →

backfilling



ideal

Support DML Statement
insert, update, delete, merge

DDL Statement
UDF → user defined function

Practise

row_count
size_of_bites
taxi_cabs
Bonus query

189
partitioning
Movie recommendation

Create function

Schema design

Normalizing data
Denormalizing data (before loading)
Shuffling data

Nested & repeated → columns
↓
improve efficiency
with relational data ←

Go-Jek 13+ PB

Normalizing data

Joins → costly

Data → repeated

Data → nested → repeated

ARRAYS → repeated

STRUCTS → pre joined table in a lump

Big query → column based

Type → Struct → Record

Arrays → regular field → Structs

Single table → STRUCTS (many)

---

nested & repeated

repeated values

Slot time compute

18 GB

7.5 GB
Bytes shuffled

, → CROSS JOIN
inside dataset (btc_value)

✓ instead joins → advantag → nested & repeated fields denormalized

✓ keep dimension table (< 10 gb) normalized

✓ Denormalize table > 10 GB