Google Cloud

Introduction to Building Batch Data Pipelines

Agenda

- EL, ELT, ETL
- Quality considerations
- How to carry out operations in BigQuery
- Shortcomings
- ETL to solve data quality issues

The method you use to load data depends on how much transformation is needed

Extract and Load

Extract, Load, Transform

Extract, Transform, Load

When would you use EL?

Architecture	**When you'd do it**
Extract data from files on Google Cloud Storage Load it into BigQuery's native storage You can trigger this from Cloud Composer, Cloud Functions, or scheduled queries	Batch load of historical data Scheduled periodic loads of log files (e.g. once a day) But only if the data are already clean and correct!

When would you use ELT?

Architecture	**When you'd do it**
Extract data from files in Google Cloud Storage into BigQuery Transform the data on the fly using BigQuery views, or store into new tables	Experimental datasets where you are not yet sure what kinds of transformations are needed to make the data useable. Any production dataset where the transformation can be expressed in SQL

Agenda

EL, ELT, ETL

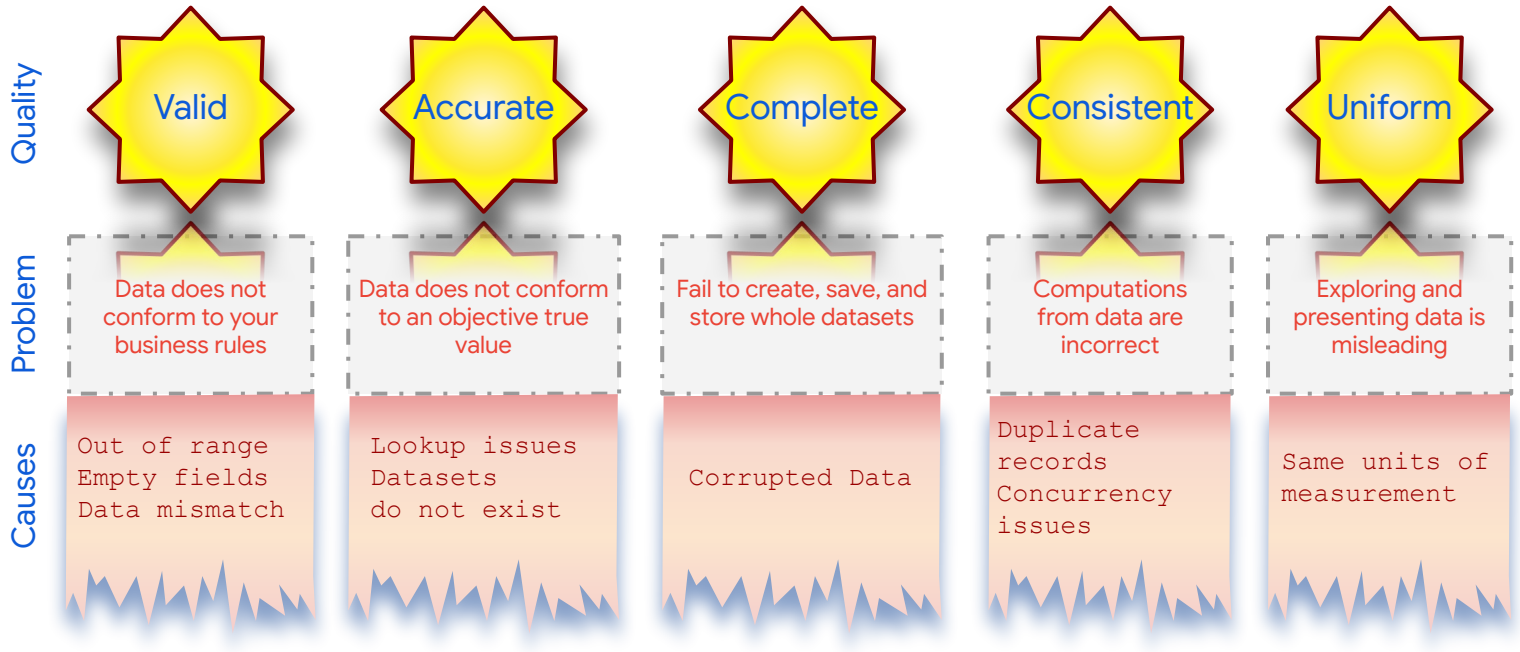
Quality considerations

How to carry out operations in BigQuery

Shortcomings

ETL to solve data quality issues

What are the purposes of Data Quality processing?



BigQuery can fix many data quality issues using SQL and Views

BigQuery

Agenda

EL, ELT, ETL

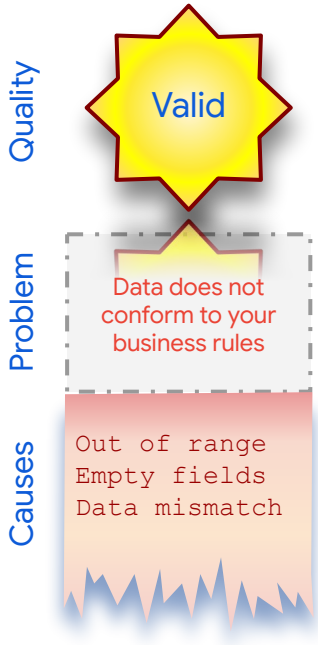
Quality considerations

How to carry out Transform operations in BigQuery

Shortcomings

ETL to solve data quality issues

Filter to identify and isolate invalid data



Setup Field Data Type Constraints

Specify fields as NULLABLE or REQUIRED

SQL: NULLABLE or REQUIRED

Proactively check for NULL values

SQL: NULL

Check and Filter for Allowable Range values SQL Conditionals:

SQL: CASE WHEN, IF ( )

Require Primary Keys / Relational Constraints in upstream source systems (remember, BigQuery is an analytics warehouse not your primary operational database)

Filter rows

```
WHERE(condition)
```

Filter aggregations

```
HAVING(condition)
```

Filters NULLs but leave blanks

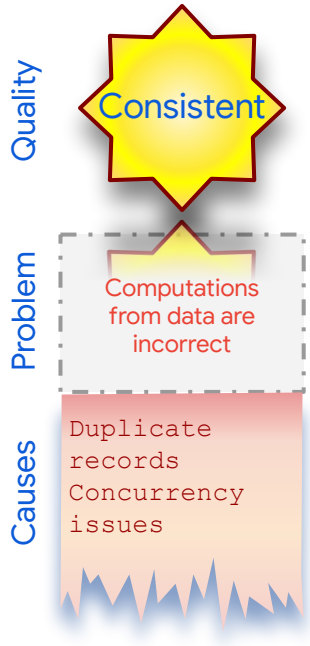
```
WHERE field IS NOT NULL
```

Filters NULLs and blanks

```
WHERE field IS NOT NULL AND field <> ""
```

A NULL is the absence of data. A BLANK is a value of data.
Consider if you are trying to filter out both NULLS and BLANKS.

Detect duplication, enforce uniqueness for consistency



Store one fact in one location and use IDs for lookup

Use String Functions to clean data:

- PARSE_DATE()
- SUBSTR()
- REPLACE()

A difference means there are duplicates

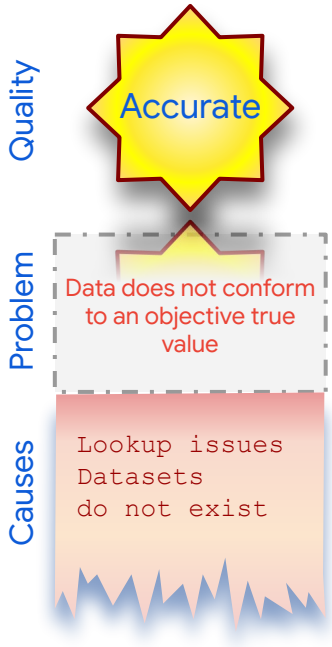
```
COUNT(DISTINCT field)
```

```
COUNT(field)
```

>1 indicates duplicates

```
COUNT(field)
GROUP BY(field)
```

Test data against known good values for accuracy



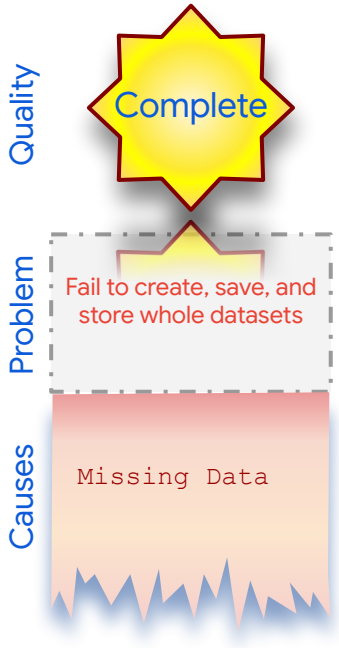
Create test cases or calculated fields to check values

SQL: (quantity_ordered * item_price) AS sub_total

Lookup values against an objective reference dataset

SQL: IN( ) with a subquery or JOIN

Identify and fill in missing values for completeness.



Thoroughly explore the existing dataset shape and skew and look for missing values

SQL: NULLIF( ) , IFNULL( ), COALESCE( )

Enrich the existing dataset with others using UNIONs and JOINs

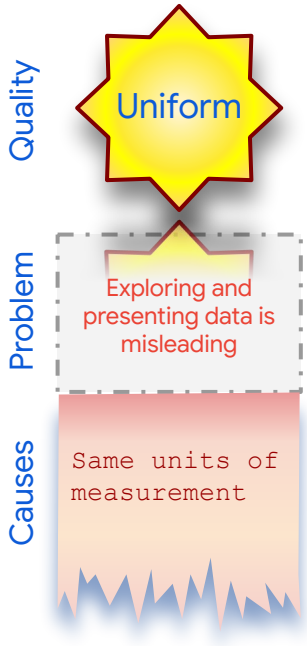
SQL: UNION, JOIN

Example: Multiple years of historical data are available for analysis

Verify file integrity with checksum values (hash, MD5)

The automatic process of detecting data drops and requesting data items to fill in the gaps is called "backfilling". It is a feature of some data transfer services.

Make data types and formats explicit for uniformity.



Document and comment your approach

Use FORMAT ( ) to clearly indicate units

SQL: FORMAT( )

CAST( ) data types to the same format and digits

SQL: CAST( )

Label all visualizations appropriately

Demo

ELT to improve data quality in BigQuery

Agenda

EL, ELT, ETL

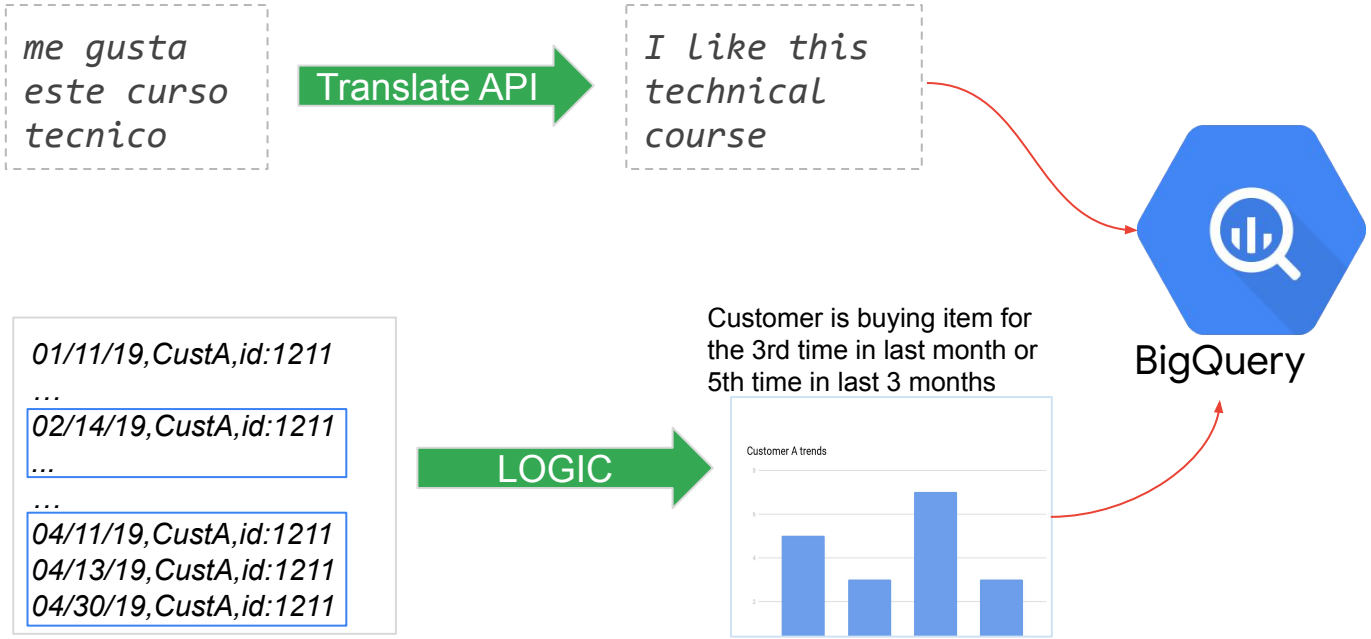
Quality considerations

How to carry out operations in BigQuery

Shortcomings

ETL to solve data quality issues

What if the transformations cannot be expressed in SQL? Or are too complex to do in SQL?



Build ETL pipelines in Dataflow and land the data in BigQuery

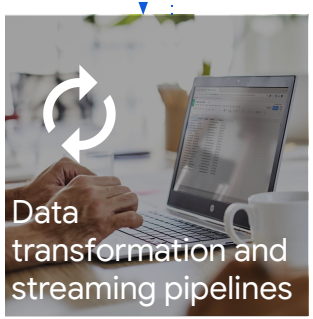
Architecture	When you'd do it
Extract data from Pub/Sub, Google Cloud Storage, Cloud Spanner, Cloud SQL, etc. Transform the data using Cloud Dataflow Have Dataflow pipeline write to BigQuery	When the raw data needs to be quality-controlled, transformed, or enriched before being loaded into BigQuery. When the data loading has to happen continuously, i.e. if the use case requires streaming. When you want to integrate with continuous integration / continuous delivery (CI/CD) systems and perform unit testing on all components.

Google Cloud offers a range of ETL tools

Cloud Pub/Sub

Data Transfer Services

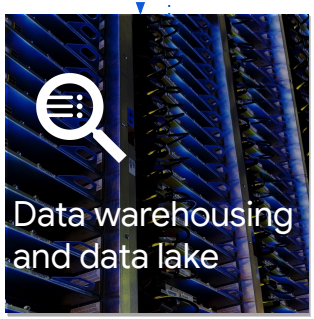
Cloud IoT Core



Cloud Dataflow

Cloud Dataproc

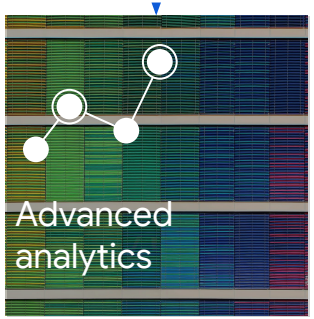
Cloud Data Fusion



BigQuery

Cloud Storage

Cloud BigTable



AI Platform

Google Data Studio

AI Platform Notebooks

looker

BI tools from Technology Partners

Google Sheets

Orchestration

Cloud Composer

Google Cloud

Agenda

EL, ELT, ETL

Quality considerations

How to carry out operations in BigQuery

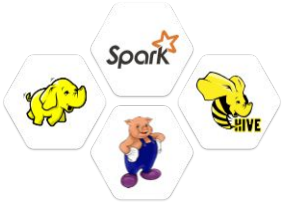
Shortcomings

ETL to solve data quality issues

Cases when you look beyond Dataflow and BigQuery

Issue	Solution
Latency, throughput	Dataflow to Bigtable
Reusing Spark pipelines	Cloud Dataproc
Need for visual pipeline building	Cloud Data Fusion

Cloud Dataproc is a managed service for batch processing, querying, streaming, and ML



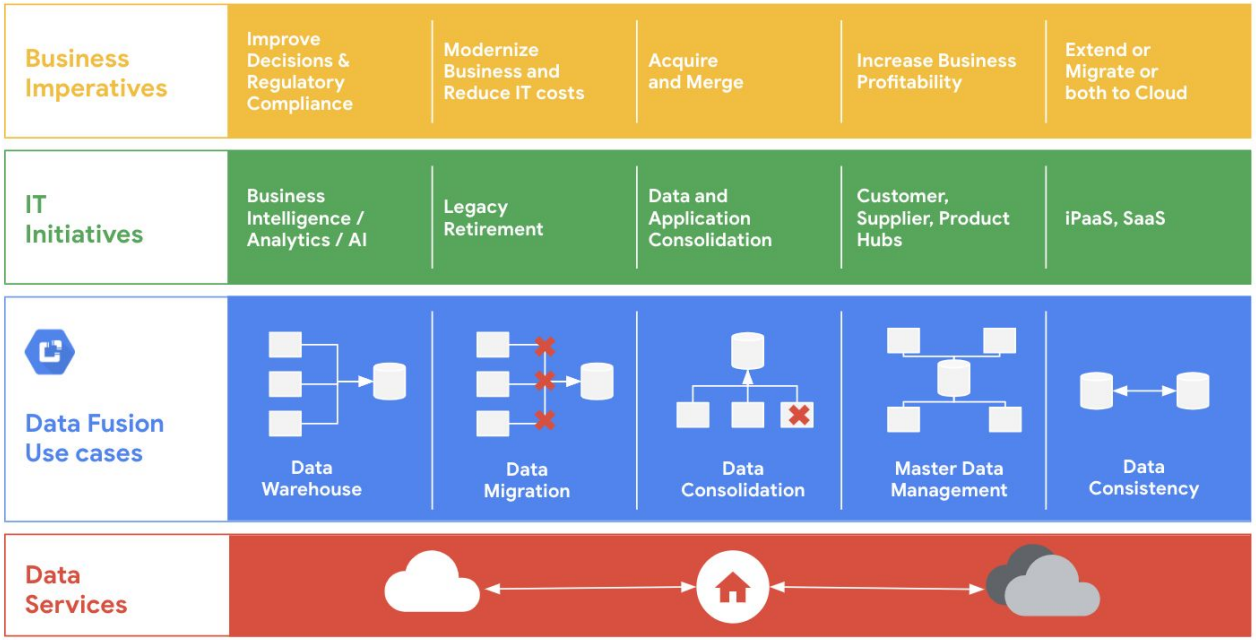
Google Cloud Platform

< Action Safe

Title Safe >

Google Cloud

Cloud Data Fusion is a fully-managed, cloud native, enterprise data integration service for quickly building and managing data pipelines



< Action Safe

Title Safe >

Google Cloud

Tracking lineage in ETL pipelines can be important

Discovery: Find the data you need

Where it came from

The processes it has been through

Its present location and condition

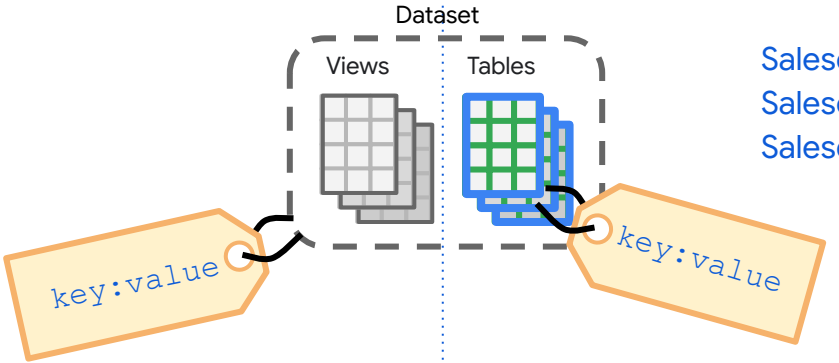
What format is it in?
What qualities does it have?
Is it fit for the intended use?
Can you transform or process it to make it fit for the intended use?

Lineage: Metadata about the data

Labels on datasets, tables, and views can help track lineage

Label

Key:	Value



Example:

A series of similar tables:

Salesdata: Europe
Salesdata: March
Salesdata: Repeat customers

< Action Safe

Title Safe >

View your datasets and labels in Data Catalog