

Welcome!



by

[Prachi Shah](#)

Software Engineer,
Engineering Leader & Mentor,
Community Volunteer.

Topic: Google BigQuery

- Fundamentals of BigQuery.
- Datasets, Tables, Views.
- Queries.
- Extract, Transfer, Load (ETL).

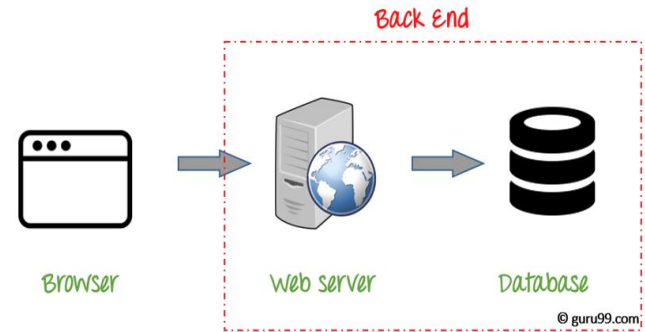
Backend Engineering

March 7, 2024

Backend Engineering

- Design, build and maintain server-side web applications

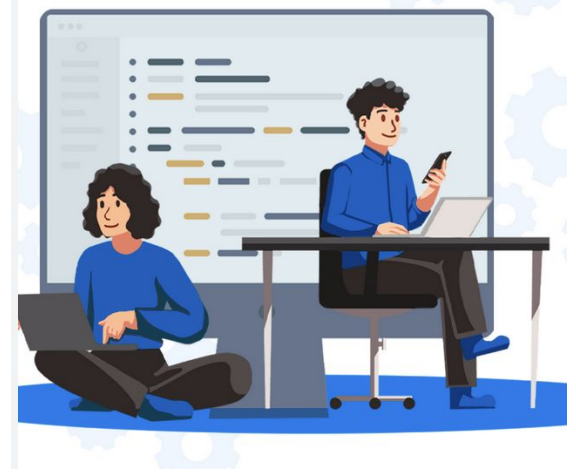
- Concepts: Client-server architecture, networking, APIs, web fundamentals, microservices, databases, security, operating systems, etc.



- Tech Stack: Java, PHP, .NET, C#, Ruby, Python, REST, AWS, Node, SQL, NoSQL, etc.

Fundamentals

- Serverless: Build and run application services without managing infrastructure.
Ex. Cloud Computing.
- Highly Scalable: Application can handle large number of users, task and data.
- Cost effective: Cost-benefit analysis and value of money.
- Multi Cloud: SaaS supporting multi cloud vendors.
Ex. MS Azure, Amazon AWS, etc.
- Data Warehouse: Central data repository for data analysis and reporting.
- Business Analytics: Transform data into insightful business decisions.



BigQuery

- Serverless Architecture: No infrastructure management, automatic scaling.
- Scalability: Analyze petabytes of data with high performance.
- Integration: Seamlessly integrates with Google Cloud Platform (GCP) and other cloud providers.
- SQL-like Querying: Familiar SQL interface for querying, data analysis and manipulation.



BigQuery



BigQuery

- Data Storage: BigQuery organizes data into datasets, each containing tables.
- Columnar Storage: Stores data in a columnar format for efficient querying.
- Pricing Model: Pay-per-query or flat-rate pricing options available.
- Security: Role-based access control (RBAC) for data protection.
- Data Formats: Supports various data formats like CSV, JSON, Avro, Parquet, etc.



```
{  
  "article_id": 3214507,  
  "article_link": "http://sample.link",  
  "published_on": "17-Sep-2020",  
  "source": "moneycontrol",  
  "article": {  
    "title": "IT stocks to see a jump this month",  
    "category": "finance",  
    "image": "http://sample.img",  
    "sentiment": "neutral"  
  }  
}
```

Datasets



- A structured collection of tables, views, and models, essential for organizing and managing access to your data in Google Cloud Platform.
- Serves as the primary mechanism for organizing data in logical groupings within a specific GCP project.
- Facilitates better data management, granular access control, and efficient analysis.
- Project Scope: Each dataset is tied to a GCP project, acting as a namespace and access control boundary.
- Data Organization: Allows for the logical grouping of tables and views, simplifying data management and analysis.

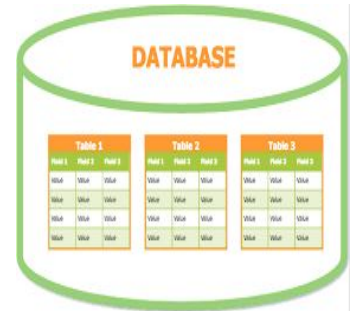
Datasets

- Data Organization: Allows for the logical grouping of tables and views, simplifying data management and analysis.
- Access Control: Permissions can be set at the dataset level, enabling specific access rights for different users or groups.
- Location Specificity: The geographic location of a dataset (region or multi-region) must be specified upon creation for performance optimization and compliance with data residency requirements.
- Example: E-commerce Analytics Dataset
 - Project: *my-ecommerce-project*
 - Dataset: *ecommerce_analytics*
 - Tables: Orders, Customers, Products, Inventory
 - Use Case: Order details, customer demographics, product performance, and inventory.



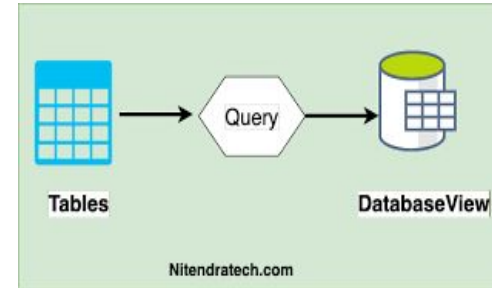
Tables

- Core components within Google BigQuery.
- Organized into datasets for structured data storage.
- Consist of rows and columns defined by a schema.
- Schema-Defined: Specifies column names and data types.
- Data Types Supported: Includes STRING, INTEGER, FLOAT, RECORD and GEOGRAPHY.
- Performance Optimization: Offers time-partitioning and clustering.
- External Tables: Enables querying external data sources.
- Example: Weather Data
 - Columns: Date, Location, Temperature, Humidity, Weather Condition.
 - Usage: Conduct climate research, analyze trends.
- Example: Product Catalog
 - Columns: ProductID, Name, Description, Price, Categories.
 - Usage: Manage product listings, analyze sales trends.



Views

- Virtual tables based on SQL queries.
 - Do not store data physically and present dynamic results.
 - Useful for simplifying complex queries and enhancing data security.
 - Dynamic Data Representation: Reflects the latest data in underlying tables.
 - Security and Access Control: Limits exposure to sensitive data.
 - Query Simplification: Encapsulates complex SQL logic.
 - Materialized Views: Improves performance by caching query results.
-
- Example: Daily Sales Summary
 - Purpose: Summarize daily sales by product.
 - Benefits: Easy access to sales insights for decision-making.
 - Top Performing Products
 - Purpose: Identify and rank top-selling products.
 - Benefits: Focuses sales and marketing efforts on high-demand items.



Queries

- Schema: Defines the structure of tables including column names and data types.
- Partitioning: Organizes data into logical partitions for improved performance.
- BigQuery supports ANSI SQL for querying data.
- Standard SQL vs. Legacy SQL: Choose between two SQL dialects.
- Query Optimization: Automatic query optimization and indexing for faster execution.
- Nested and Repeated Fields: Handle complex data structures efficiently.
- Supports a variety of operations, like SELECT statements, aggregate functions, JOIN, etc.
- User-Defined Functions (UDFs): Extend query support for custom data manipulation functions.



Queries

- Purpose: Retrieve all records from a table.

```
SELECT * FROM my_dataset.my_table;
```

- Basic operation to fetch all data from a specified table.



- Purpose: Join/Merge data from two tables based on a related column.

```
SELECT orders.OrderID, customers.CustomerName FROM orders JOIN  
customers ON orders.CustomerID = customers.CustomerID;
```

- Combine related data from different tables for comprehensive analysis.

- Purpose: Calculate the total sales by product.

```
SELECT ProductID, SUM(Amount) AS TotalSales FROM sales_table  
GROUP BY ProductID;
```

- Summarize sales data to understand product performance.

ETL

Extract, Transform, Load (ETL):

- A process used in data warehousing to move data from multiple sources into a single, centralized database, data warehouse, or data lake.
- Extract: Retrieve data from various sources like Google Cloud Storage, Google Drive, etc.
- Transform: Cleanse, enrich, and transform data using SQL queries.
- Load: Load processed data into BigQuery tables for analysis.
- Data Transfer Service: Automates data movement from external sources to BigQuery.
- Dataflow Integration: Utilize Google Dataflow for complex ETL workflows.
- BigQuery automates much of the ETL process, making it faster and more efficient.
- Enables businesses to efficiently analyze large datasets, derive insights, and make data-driven decisions.



ETL

- Example: Analyzing E-commerce Data

- Extract: Retrieve sales data from Google Cloud Storage.

```
SELECT * FROM project.dataset.sales WHERE date BETWEEN '2024-01-01' AND '2024-01-31';
```

- Transform: Calculate revenue, analyze customer behavior.

```
SELECT customer_id, SUM(amount) AS total_spent FROM project.dataset.sales GROUP BY customer_id;
```



- Load: Load aggregated data into BigQuery for visualization.

```
CREATE TABLE project.dataset.customer_spending AS SELECT customer_id, SUM(amount) AS total_spent FROM project.dataset.sales GROUP BY customer_id;
```

ETL

- Example: Real-time Analytics

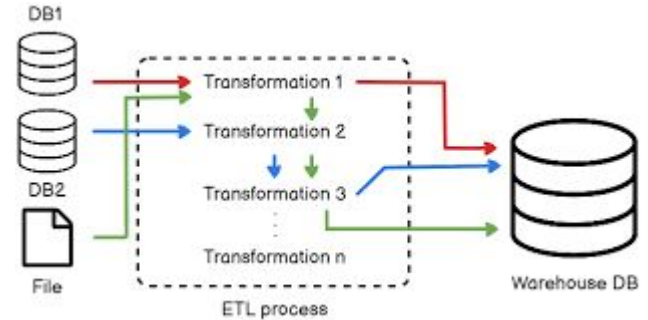
- Streaming: Ingest streaming data using BigQuery Streaming API.

```
INSERT INTO project.dataset.stream_data (timestamp, event_type, data)
VALUES (CURRENT_TIMESTAMP(), 'click', '{ "user_id": "123",
"page": "/product", "action": "click" }');
```

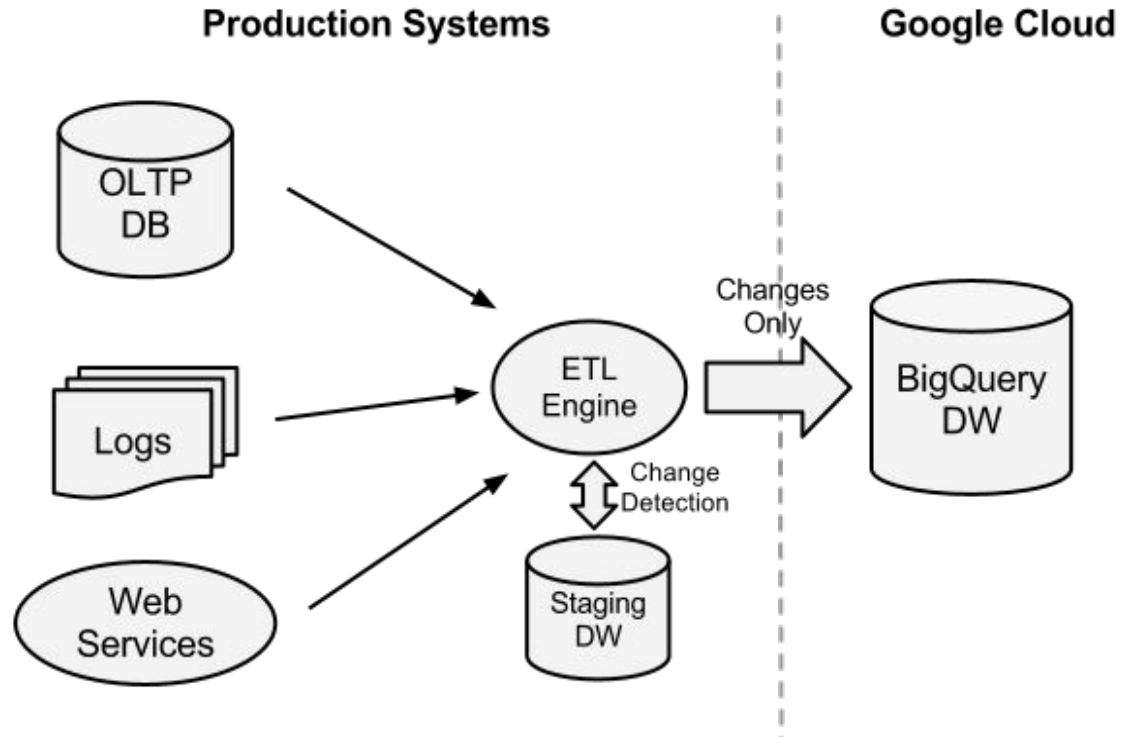
- Transform: Process and analyze streaming data in real-time.

```
SELECT event_type, COUNT(*) FROM
project.dataset.stream_data
GROUP BY event_type;
```

- Load: Persist processed data in BigQuery tables for further analysis.



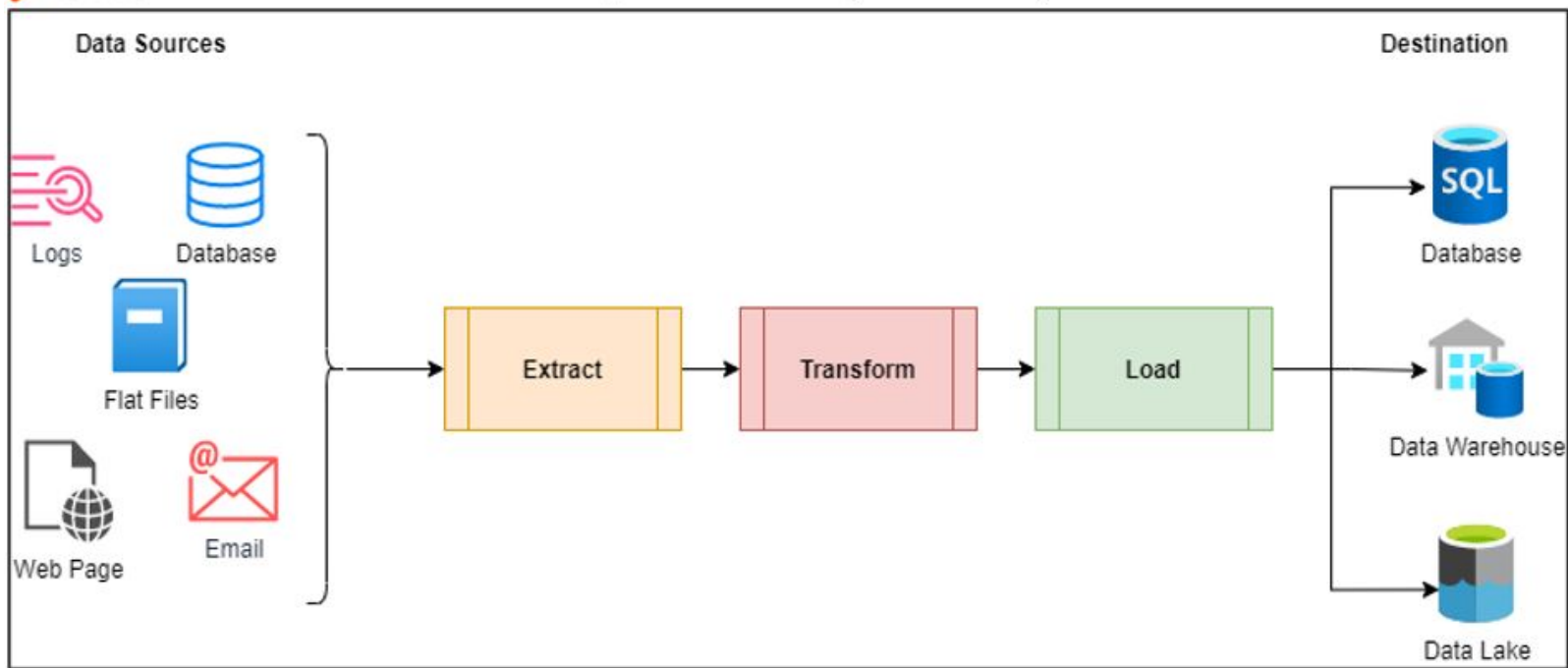
ETL



ETL



The ETL Pipeline: Extract, Transform, Load



Console

Google Cloud Platform

Project for Coupler

Search products and resources

FEATURES & INFO

SHORTCUT

HIDE PREVIEW FEATURES

Explorer

+ ADD DATA

Type to search

?

Viewing pinned projects.

project-for-coupler

Applicants

Applicants

feisty-audio-282807

Applicants

SHARE TABLE

COPY TABLE

DELETE TABLE

EXPORT

Schema

Details

Preview

Row	id	Position	Application_Date	Stage_Name	Applicant_Status	Recruiter_Name	Country
1	199	Recruiter	2019-10-07	RPI	lost	Howard Wolowitz	United Kingdom
2	211	Recruiter	2019-11-21	RPI	open	Leslie Winkle	Philippines
3	263	Recruiter	2020-02-04	RPI	won	Sheldon Cooper	Colombia
4	272	Recruiter	2020-04-02	RPI	lost	Raj Koothrappali	Afghanistan
5	323	Recruiter	2020-02-17	RPI	won	Howard Wolowitz	China
6	374	Recruiter	2019-10-04	RPI	lost	Leslie Winkle	Russia
7	376	Recruiter	2020-05-05	RPI	won	Leslie Winkle	Russia
8	389	Recruiter	2019-09-08	RPI	lost	Sheldon Cooper	Mongolia
9	401	Recruiter	2020-02-25	RPI	open	Leslie Winkle	Belarus
10	494	Recruiter	2020-04-09	RPI	lost	Howard Wolowitz	Norway

Rows per page:

100

1 - 100 of 1000

First page

>| Last page

JOB HISTORY

QUERY HISTORY

SAVED QUERIES

Copyright © 2024 by [Prachi Shah](#)

Console

Google Cloud MyProject1 Search (/) for resources, docs, products, and more Search

BigQuery Explorer + ADD <

Analysis

- BigQuery Studio
- Data transfers
- Scheduled queries
- Analytics Hub
- Dataform
- Partner Center

Migration

- Assessment
- SQL translation

Administration

- Capacity management
- BI Engine
- Policy tags

Release Notes

Viewing starred resources. SHOW STARRED ONLY

- myproject-295313
- myproject1-381000
 - External connections
 - cloud_monitoring_import...
 - dataset2
 - mydataset
 - 12345
 - MyTable12345
 - asdgy
 - remote_function_test

SUMMARY

Nothing currently selected

Untitled 2 RUN SAVE SHARE SCHEDULE MORE

```
1 SELECT
2   word,
3   SUM(word_count) AS count
4 FROM
5   `bigquery-public-data`.samples.shakespeare
6 WHERE
7   word LIKE "%raisin"
8 GROUP BY
9   word
10
11
```

Console

Google Cloud Platform

Start BigQuery

Search products and resources

1

BigQuery

FEATURES & INFO

SHORTCUT

Query history

Saved queries

Job history

Transfers

Scheduled queries

Reservations

BI Engine

Resources

Search for your tables ...

start-bigquery-294922

start_bigquery

brooklyn_br...

bigquery-public-data

data-pipeline-292113

Query editor

+ COMPOSE NEW QUERY

HIDE EDITOR

FULL SCREEN

```
1 SELECT
2 *
3 FROM
4 `start-bigquery-294922.start_bigquery.brooklyn_bridge_pedestrians`
5 LIMIT
6 10
```

Processing location: US

Run

Save query

Save view

Schedule query

More

This query will process 0 B when run.

Query results

SAVE RESULTS

EXPLORE DATA

Query complete (2.7 sec elapsed, 2.4 MB processed)

Job information

Results

JSON

Execution details

Row	hour_beginning	location	Pedestrians	Towards_Manhattan	Towards_Brooklyn	weather_summary	temperature
1	2017-10-01 00:00:00 UTC	Brooklyn Bridge	44	30	14	clear-night	52
2	2017-10-01 00:59:59.999999 UTC	Brooklyn Bridge	30	17	13	partly-cloudy-night	53
3	2017-10-01 02:00:00 UTC	Brooklyn Bridge	25	13	12	partly-cloudy-night	52
4	2017-10-01 03:00:00 UTC	Brooklyn Bridge	20	11	9	partly-cloudy-night	51

Summary

- Skill set:
 - Database.
 - SQL and complex querying.
 - ETL (Extract, Transform, Load).
 - Contextual knowledge (business domain knowledge).
- Applications:
 - Provides machine learning capabilities.
 - Supports predictive analysis.
 - Perform business analytics.
 - Conduct geospatial analysis and computer vision.
- Features:
 - Supports upto 20,000 tables.
 - Can have an export file size of upto 1 GB.
 - In-memory caching.

