Welcome!

WWCode San Francisco - Backend Study Group

March 7, 2024

- We'll start in a moment :)
- We are **RECORDING** tonight's event
- We may plan to take screenshots for social media
- If you are comfortable, turn the video ON. If you want to be anonymous, then turn the video off
- We'll make some time for Q&A at the end of the presentation
- Feel free to take notes
- Online event best practices:
    - Don't multitask. Distractions reduce your ability to remember concepts
    - Mute yourself when you aren't talking
    - We want the session to be interactive
    - Use the 'Raise Hand' feature to ask questions
- **By attending our events, you agree to comply with our Code of Conduct**

WOMEN WHO CODE /san-francisco

Introduction & Agenda

- Welcome from WWCode!
- Our mission: Empower women to excel in technology careers.
- At Backend Study Group, we learn and discuss backend engineering concepts.

Prachi Shah
Instructor
Sr. Software Engineer, Unity
Director, WWCode SF

Anjali Bajaj
Host,
Lead, WWCode SF

Google BigQuery:

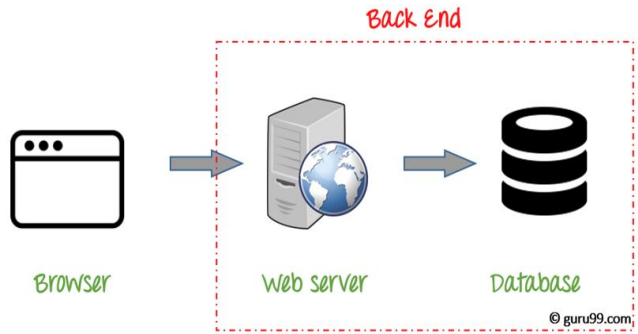
- Fundamentals of BigQuery.
- Datasets, Tables, Views.
- Queries.
- Extract, Transfer, Load (ETL).
- Q & A.

WOMEN WHO CODE /san-francisco

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

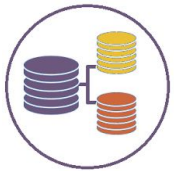
- Data Storage: BigQuery organizes data into datasets, each containing tables.
- Columnar Storage: Stores data in a columnar format for efficient querying.
- Pricing Model: Pay-per-query or flat-rate pricing options available.
- Security: Role-based access control (RBAC) for data protection.
- Data Formats: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

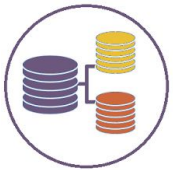
• Data Organization: Allows for the logical grouping of tables and views, simplifying data management and analysis.

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.

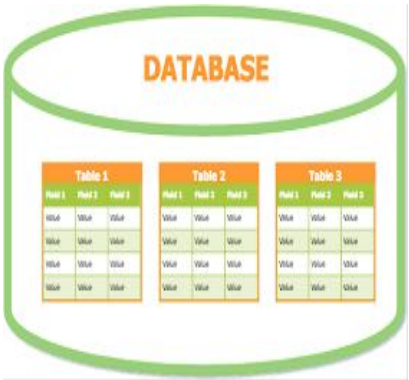
• Location Specificity: The geographic location of a dataset (region or multi-region) must be specified upon creation for performance optimization and compliance with data residency requirements.



• 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

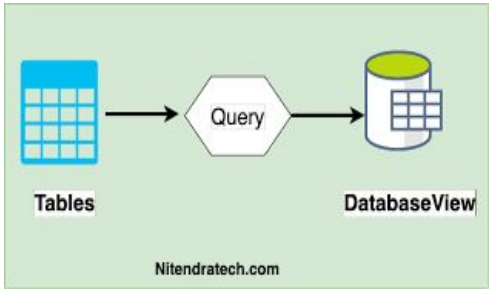
- Core components within Google BigQuery.
- Organized into datasets for structured data storage.
- Consist of rows and columns defined by a schema.
- Schema-Defined: Specifies column names and data types.
- Data Types Supported: Includes STRING, INTEGER, FLOAT, RECORD and GEOGRAPHY.
- Performance Optimization: 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.

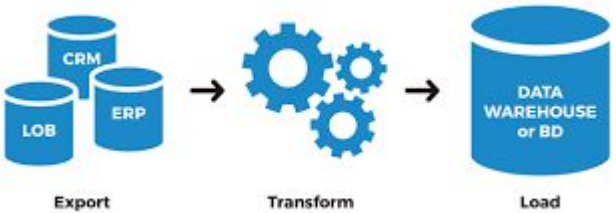
- Purpose: Calculate the total sales by product.

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

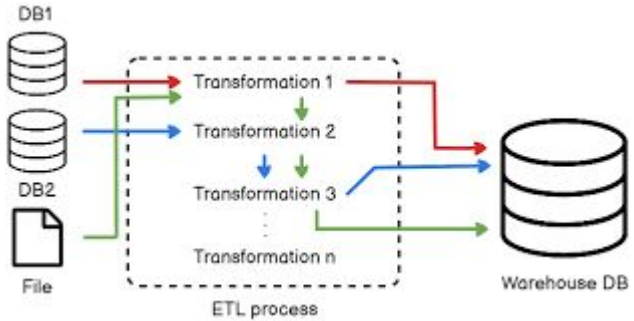
ETL

- 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" }');
```

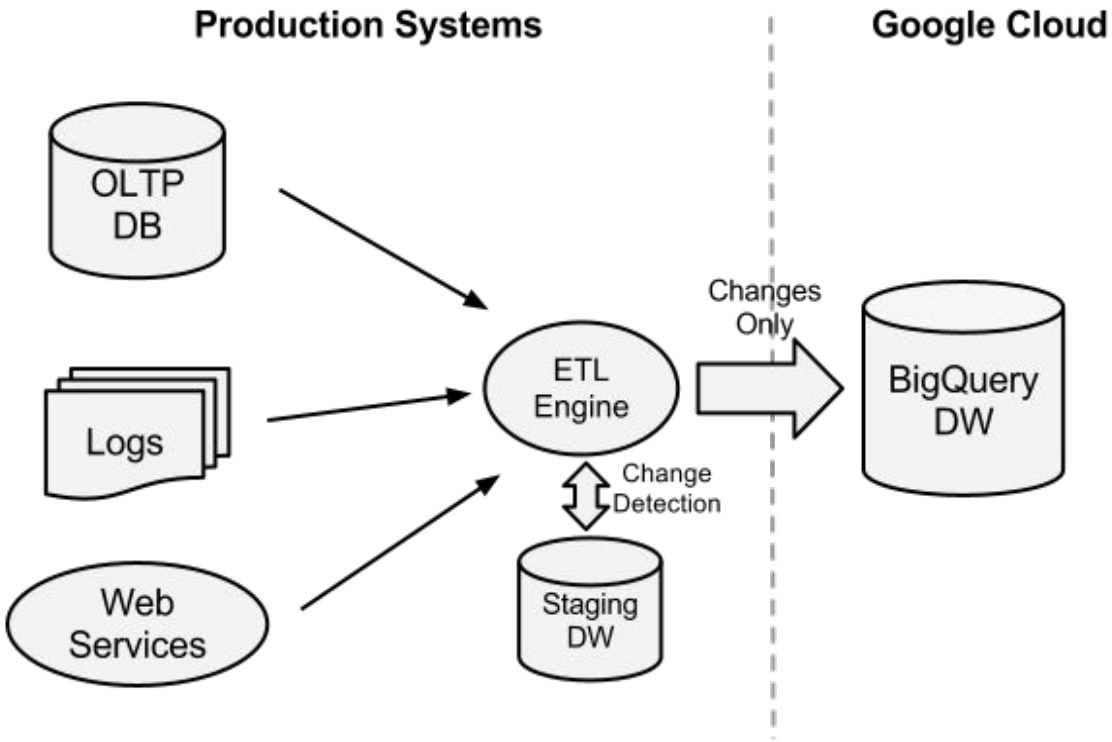
  - Transform: Process and analyze streaming data in real-time.

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

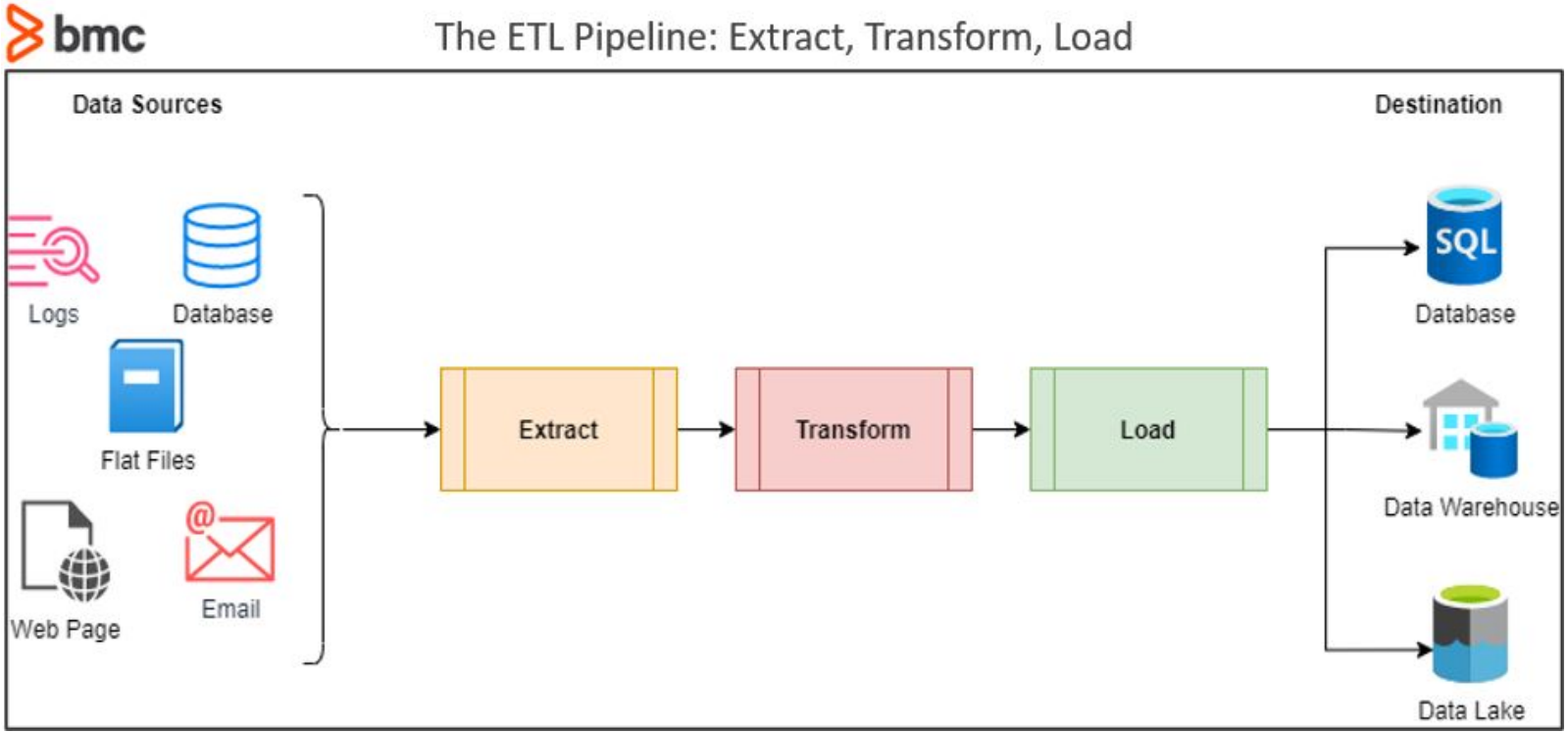


  - Load: Persist processed data in BigQuery tables for further analysis.

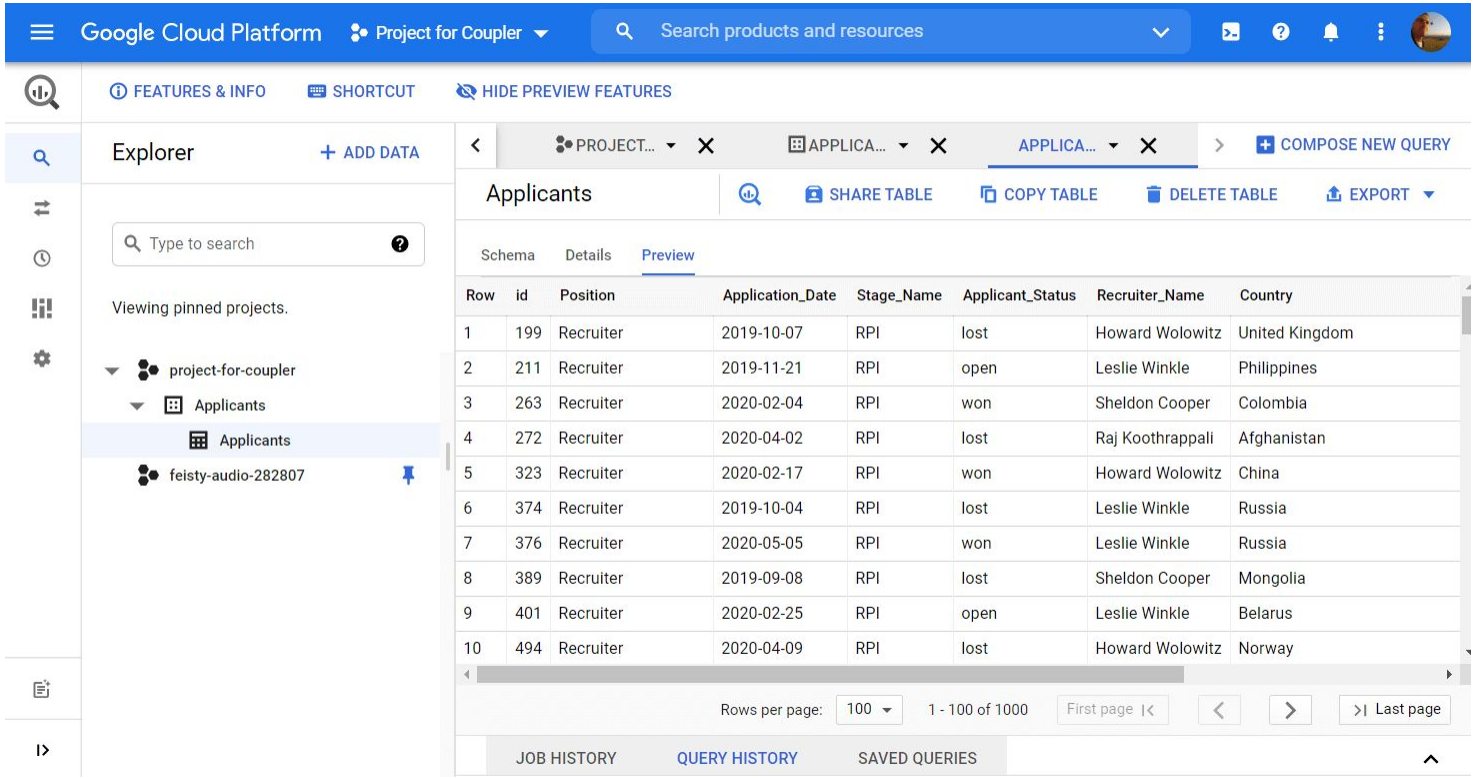
ETL



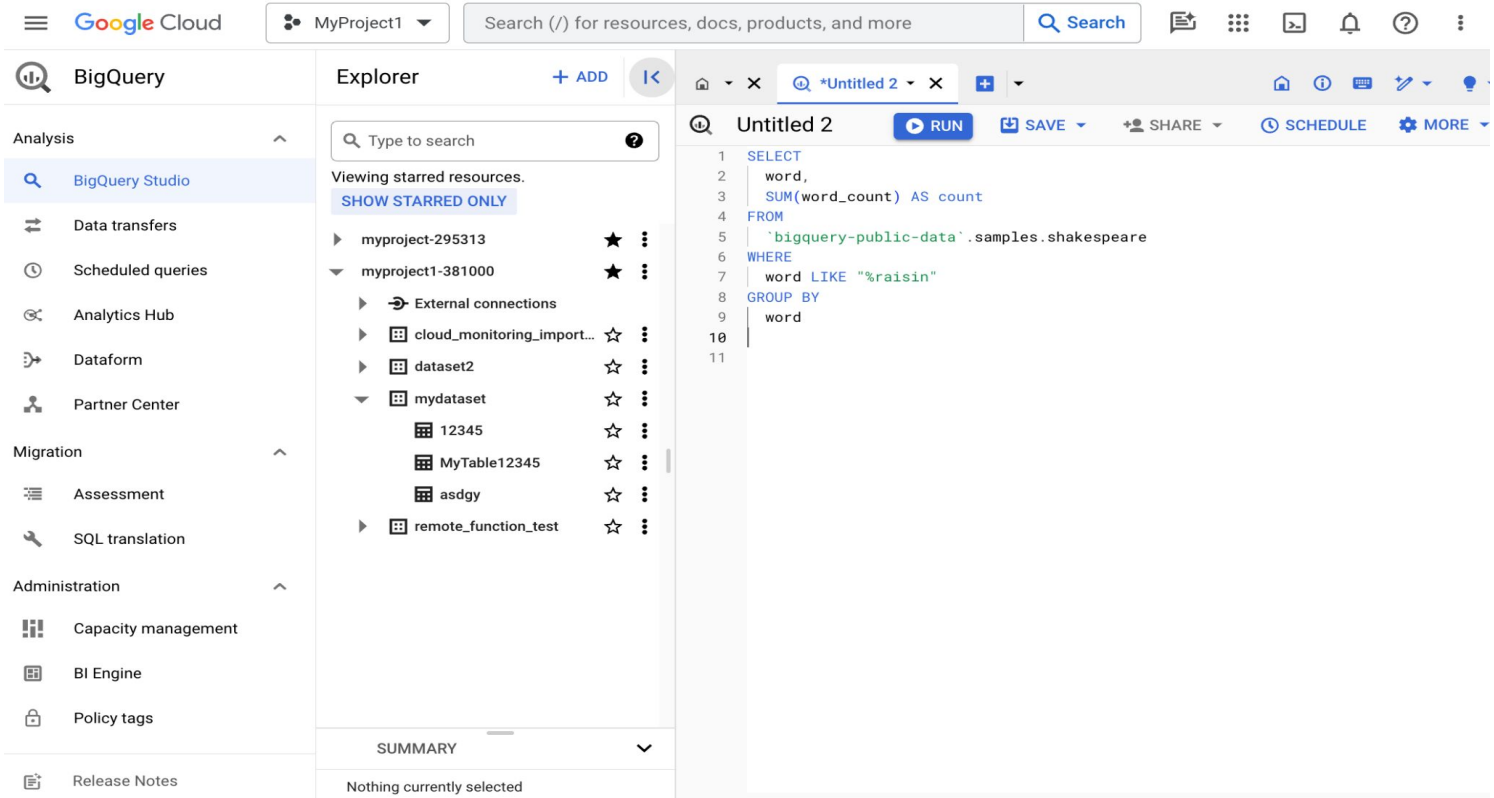
ETL



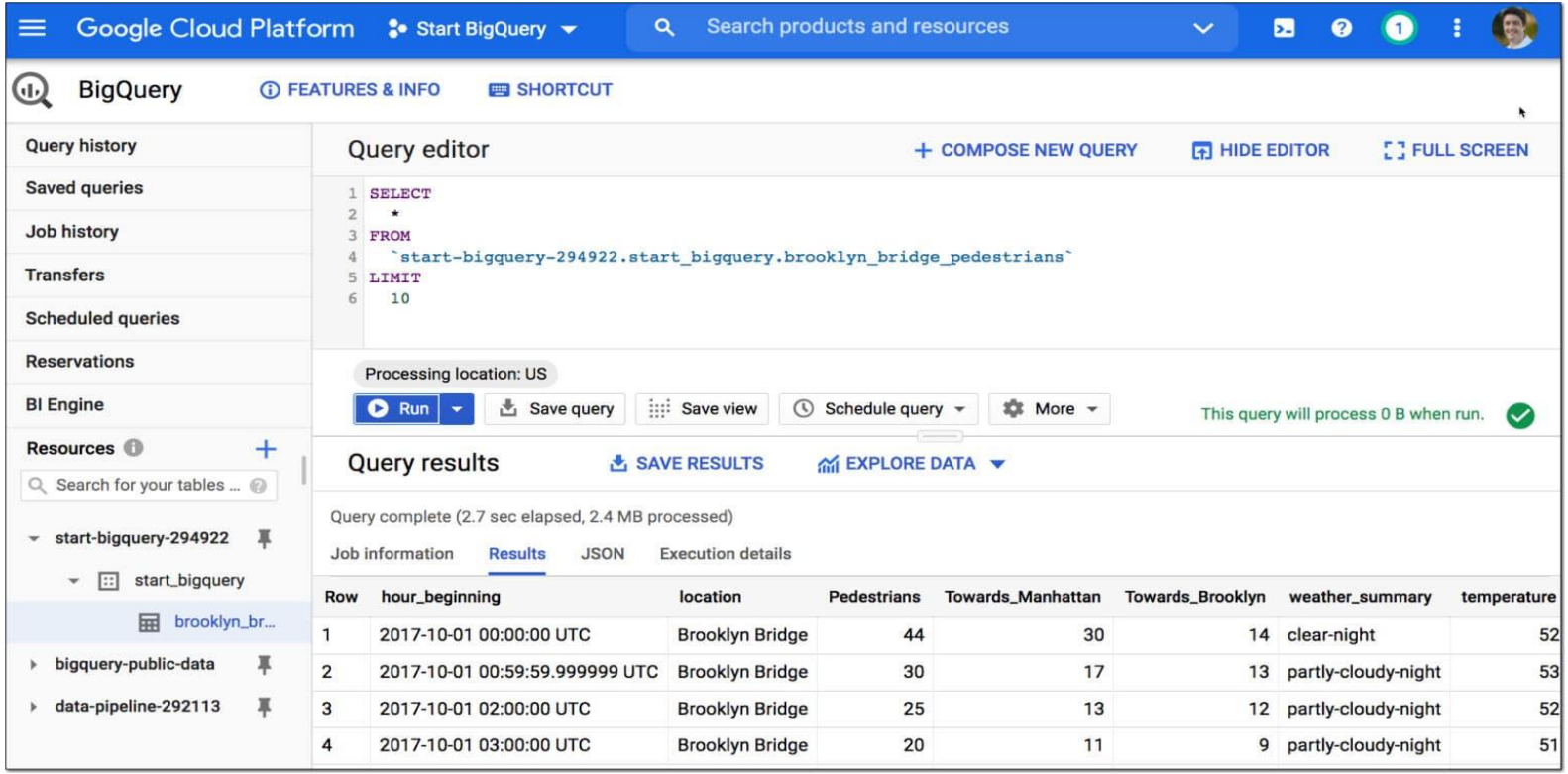
Console



Console



Console



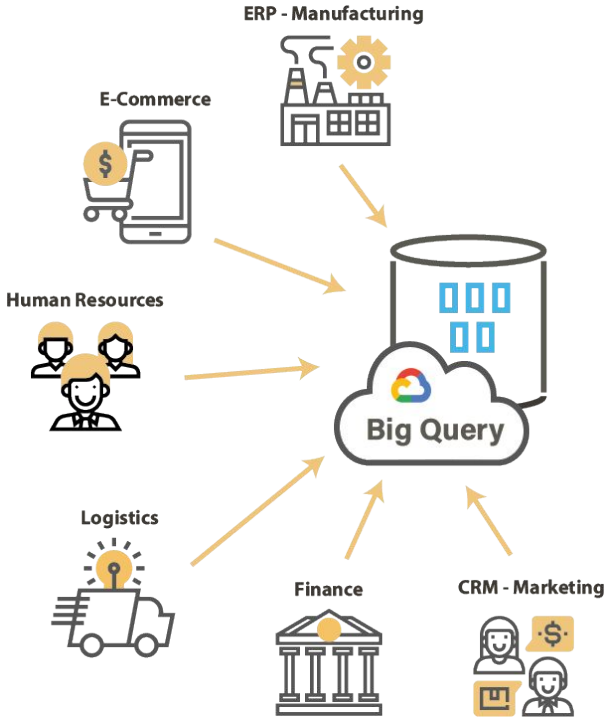
Row	hour_beginning	location	Pedestrians	Towards_Manhattan	Towards_Brooklyn	weather_summary	temperature
1	2017-10-01 00:00:00 UTC	Brooklyn Bridge	44	30	14	clear-night	52
2	2017-10-01 00:59:59.999999 UTC	Brooklyn Bridge	30	17	13	partly-cloudy-night	53
3	2017-10-01 02:00:00 UTC	Brooklyn Bridge	25	13	12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



Backend Study Group

References:

- BigQuery Storage
- BigQuery Architecture

Backend Study Group:

- Presentations on GitHub and session recordings available on WWCode YouTube channel
- More events to come! RSVP at meetup.com/women-who-code-sf

Women Who Code:

- Technical Tracks and Digital Events for more events
- Join the Digital mailing list for updates about WWCode
- Contacts us at: contact@womenwhocode.com
- Join our Slack workspace and join *#backend-study-group*!

You can unmute and talk or use the chat

WOMEN WHO CODE /san-francisco