GCP BigQuery Tutorial

BigQuery Nedir ?
BigQuery Mimarisi Nedir Nasıldır ?

GCP BigQuery Konsol özellikleri
SQL ile BigQuery Farklılıkları nedir ?

BigQuery Nedir

Google Cloud BigQuery, büyük ölçekte veri analizi ve sorgulama yapmak için kullanılan yönetilen bir veri ambarı ve analitik veritabanı hizmetidir. BigQuery, Google Cloud'un büyük veri ve veri analizi alanındaki önemli hizmetlerinden biridir ve birçok özellik sunar:

1. **Hızlı ve Ölçeklenebilir Veri Analizi:** BigQuery, paralel sorgulama ve dağıtılmış hesaplama gücü sayesinde büyük veri kümelerini hızlı bir şekilde analiz etme yeteneği sunar. Verilerinizi saniyeler içinde sorgulayabilir ve sonuçlarını alabilirsiniz.
2. **Yönetilmeyen Hizmet:** BigQuery, altyapı yönetimiyle ilgilenmenize gerek kalmadan kullanıma hazır bir hizmet sunar. Veri depolama, sorgulama performansı optimizasyonu ve yedekleme gibi karmaşık operasyonlar otomatik olarak yönetilir.
3. **Geniş Veri Formatı Desteği:** BigQuery, CSV, JSON, Avro gibi farklı veri formatlarını destekler. Ayrıca Parquet ve ORC gibi sıkıştırılmış ve sütun tabanlı veri formatlarını da destekler.
4. **Federasyon Desteği:** BigQuery, harici veritabanlarına ve veri kaynaklarına (örneğin Google Sheets) federatif sorgulama yeteneği sunar. Bu, farklı kaynaklardaki verileri tek bir sorgu altında birleştirme kolaylığı sağlar.
5. **Kullanıcı Dostu Arabirim:** BigQuery, web tabanlı kullanıcı arayüzü veya komut satırı araçları ile kolayca kullanılabilir. Veri kümesi oluşturma, sorgulama çalıştırma ve sonuçları görselleştirme işlemleri kullanıcı dostu bir şekilde yapılabilir.
6. **Veri Paylaşımı ve İşbirliği:** BigQuery, verilerinizi diğer kullanıcılar veya hizmetler ile paylaşmanıza olanak tanır. Veri paylaşımını yönetebilir ve gerektiğinde güvenlik ayarları uygulayabilirsiniz.
7. **Entegrasyon ve Analitik Araçlarla Uyum:** BigQuery, popüler veri analitik araçları ile entegrasyon sağlar. Bu sayede verilerinizi çeşitli analitik ve görselleştirme araçları ile işleyebilirsiniz.
8. **Veri Güvenliği ve İzleme:** BigQuery, veri güvenliği konusunda önemli özellikler sunar. Veriler şifrelenir ve kimlik doğrulama, yetkilendirme ve erişim kontrolleri sağlar.
9. **Ücretlendirme Modeli:** BigQuery, kullanımına göre ücretlendirilen bir hizmettir. Depolama ve sorgulama işlemleri için ayrı ayrı ücretlendirilir.

Google Cloud BigQuery, büyük veri analizi yapmak isteyen işletmeler ve veri bilimciler için güçlü bir araçtır ve çeşitli özellikleri sayesinde veri analizi süreçlerini daha hızlı ve etkili bir şekilde gerçekleştirmelerine yardımcı olur.

BigQuery Columnar Database

Columnar database, veri depolama ve yönetimini daha verimli hale getiren bir veritabanı türüdür. Geleneksel satır tabanlı (row-based) veritabanlarına karşı bir alternatif olarak ortaya çıkmıştır. Temel farkı, veriyi sütunlara göre depolama ve işleme yaklaşımını benimsemesidir.

Geleneksel satır tabanlı veritabanlarında, veri satırlar halinde depolanır ve her bir satır bir kaydı temsil eder. Bu tür veritabanlarında genellikle birçok sütun içeren kayıtlar bulunur. Bu yaklaşım, işlem odaklı uygulamalar için uygundur.

Buna karşılık, columnar database'de veriler sütunlar halinde depolanır. Her bir sütun, belirli bir veri tipini temsil eder. Bu yaklaşım, özellikle analitik işlemler ve veri sorgulama için daha etkili bir performans sağlar. Columnar database'lerin bazı temel özellikleri şunlar olabilir:

1. **Sıkıştırma Avantajı:** Aynı türdeki verilerin bir araya gelmesi, sıkıştırma algoritmalarının daha etkili çalışmasına olanak tanır. Bu da daha az depolama alanı kullanılmasını sağlar.
2. **Veri Sorgulama Performansı:** Veriler sütunlar halinde depolandığı için, sorgulama işlemleri genellikle daha hızlı gerçekleştirilir. Özellikle büyük veri küplerini analiz etmek için kullanıldığında performans avantajı sağlar.
3. **Sadece İhtiyaç Duyulan Sütunlar İşlenir:** Sorgular, yalnızca ilgili sütunları işlemek için tasarlanmıştır. Bu da gereksiz veri taşıma işlemlerini azaltır ve işlem sürelerini kısaltır.
4. **Paralel İşleme:** Columnar database'ler, paralel işleme yapısına uygun olarak tasarlanır. Bu, çoklu çekirdekli işlemcilerin ve dağıtılmış sistemlerin etkin kullanımına olanak tanır.
5. **Analitik ve Veri Madenciliği Uygulamaları:** Columnar database'ler genellikle büyük veri küplerini analiz etmek, veri madenciliği yapmak ve iş zekası uygulamaları oluşturmak için kullanılır.

BigQuery Mimarisi

Google Cloud BigQuery, büyük ölçekte veri analizi yapmayı sağlayan hızlı, ölçeklenebilir ve yönetilen bir veri analizi hizmetidir. BigQuery'nin mimarisi, veri analizi gereksinimlerini karşılayacak şekilde tasarlanmıştır ve temelde üç ana bileşenden oluşur:

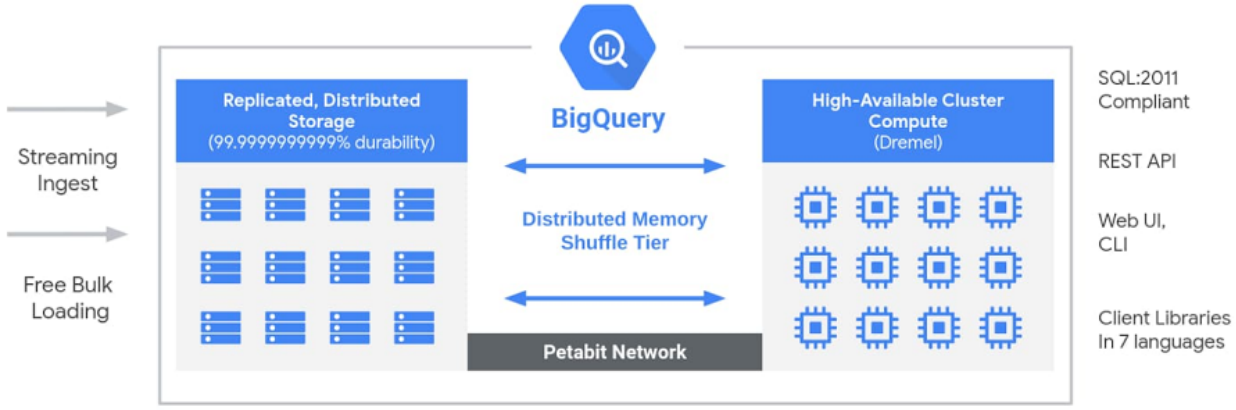
1. **Storage (Depolama) Tabakası:**
    - BigQuery, verileri sütun tabanlı bir depolama modelinde saklar. Bu, analitik sorgulamalar için daha etkili bir performans sağlar.
    - Veriler Google Cloud Storage'da depolanır. Bu sayede BigQuery, veriyi fiziksel olarak sütunlar halinde gruplandırabilir ve yönetebilir.

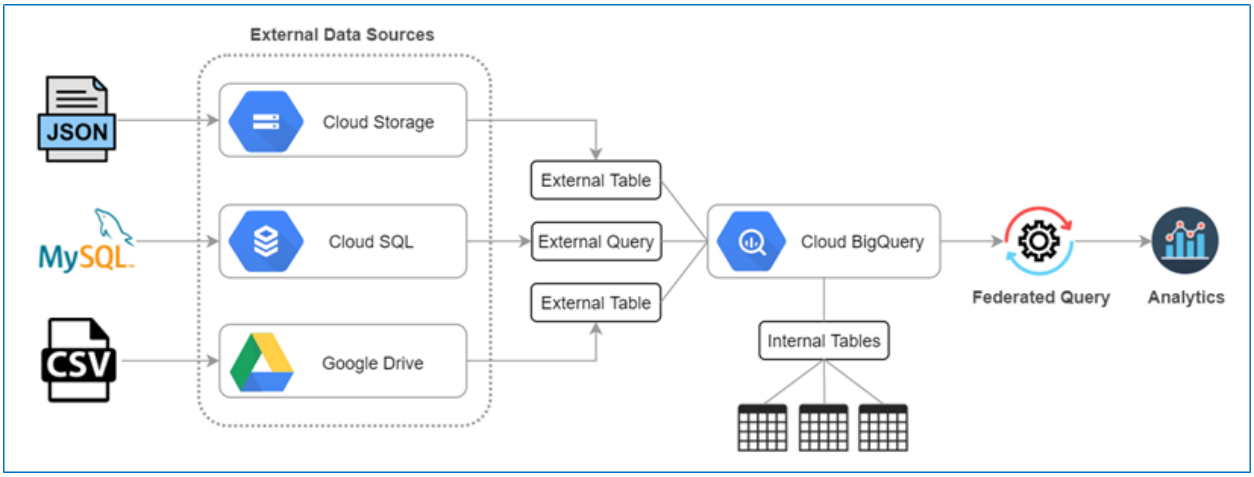
- BigQuery, sıkıştırma ve kodlamayı otomatik olarak uygulayarak veri depolama maliyetlerini optimize eder.

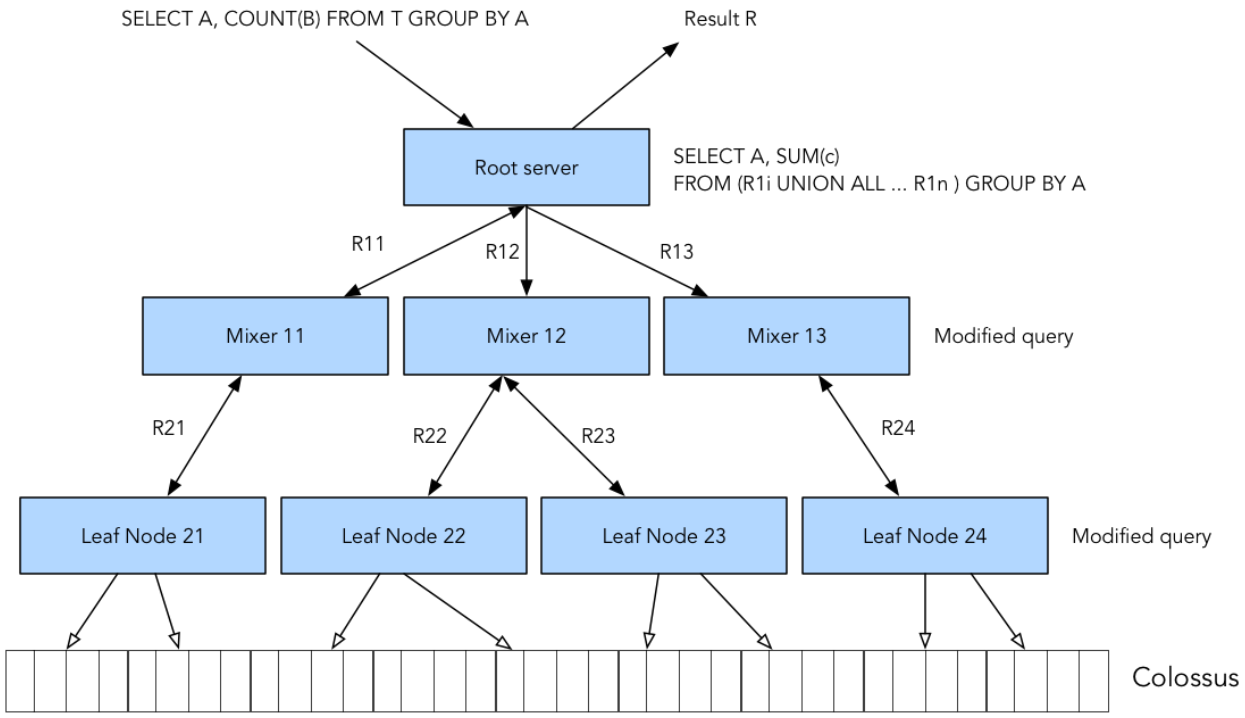
2. **Query Engine (Sorgu Motoru) Tabakası:**
   - Bu tabaka, yürütülen sorguları işlemek ve sonuçları döndürmek için kullanılır.
   - Paralel sorgulama ve dağıtılmış işlem yapısı sayesinde hızlı ve ölçeklenebilir sorgulama performansı sağlar.
   - Verilerin fiziksel dağılımı ve optimizasyonları bu tabakada gerçekleşir.
3. **Execution Control (Yürütme Kontrol) Tabakası:**
   - Bu tabaka, sorgu yürütme ve kaynak yönetimi işlevlerini üstlenir.
   - Sorgu yürütme sırasında otomatik paralellik, veri bölümlendirme ve sorgu planlaması gibi görevleri yönetir.
   - Kaynak yönetimi, sorgu performansını optimize etmek ve bütçeyi kontrol altında tutmak için gereken kontrolleri sağlar.

BigQuery'nin mantığı, veriyi depolama tabakasında sütunlar halinde saklamak ve sorgu motoru tabakasında paralel işleme ile hızlı ve ölçeklenebilir sorgular yürütmek şeklinde çalışır. Kullanıcılar sorgularını gönderdiklerinde, BigQuery otomatik olarak sorguları paralel olarak dağıtarak verileri işler ve sonuçları döndürür. Bu mimari, büyük veri kümelerini hızlı ve etkili bir şekilde analiz etmek için tasarlanmıştır.

Google Cloud BigQuery, veri analizi ihtiyaçlarını karşılamak için gereken ölçeklenebilirlik, hız ve yönetim kolaylığını sağlayan bir bulut tabanlı hizmettir.







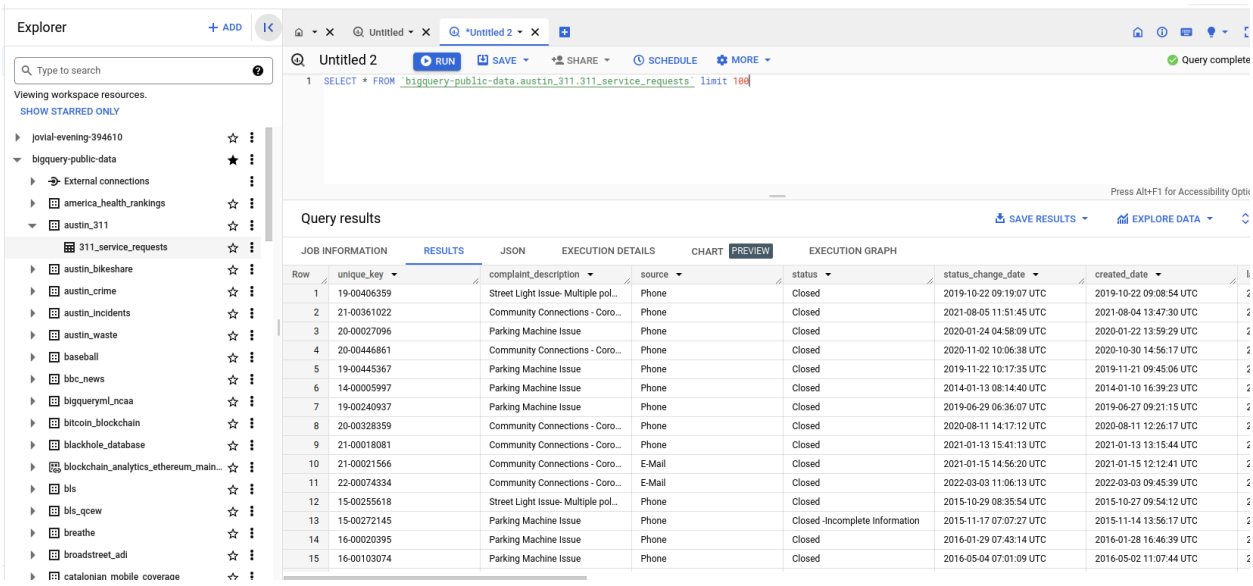
GCP BigQuery-SQL Query

Öncelikle örnek bir veri seti yüklemeliyiz. Bunun farklı yolları mevcuttur. BigQuery ekranına geldikten sonra ADD tıkladıktan sonra açılan ekrandan kullanacağınız örnek çalışmalarınızı yükleyebilirsiniz. Örnek olması açısından biz Proje adı ile kendi içinde yer alan örnek veri setlerini import edeceğiz.

ADD >> start a project by name >> bigquery-public-data

İçeri aktardığımız örnek veri setlerinden birinden örnek olarak bir sql select sorgusu oluşturalım.

Burada farklı olarak table alışık olduğumuz şekilde table ismi vermek yerine ***proje_id.dateset_adı.tablo_adı şeklinde olmasıdır.***




```
SELECT * FROM `bigquery-public-data.austin_311.311_service_requests` limit 100
```

Örnek Shell Komutları

1. **Bir Veritabanı Örneği Oluşturma:**

```
bq mk --dataset PROJECT_ID:DATASET_NAME
```

2. **Veri Kümelerini Listeleme:**

```
bq ls PROJECT_ID
```

3. **Bir Veri Setinin Yapısını Görüntüleme:**

```
bq show PROJECT_ID:DATASET_NAME
```

4. **Bir SQL Sorgusu Çalıştırma ve Sonuçları Görüntüleme:**

```
bq query 'SELECT * FROM `PROJECT_ID.DATASET_NAME.TABLE_NAME`'
```

5. **Veri Setine Dosya İçeriği Aktarma:**

```
bq load --source_format=CSV PROJECT_ID:DATASET_NAME.TABLE_NAME PATH_TO_CSV_FILE SCHEMA
```

6. **Dosya İçeriğini Veri Setine Aktarma (Avro Formatı):**

```
bq load --source_format=AVRO PROJECT_ID:DATASET_NAME.TABLE_NAME PATH_TO_AVRO_FILE SCHEMA
```

7. **Bir Tabloyu Dışa Aktarma:**

```
bq extract PROJECT_ID:DATASET_NAME.TABLE_NAME DESTINATION_FILE
```

8. **Bir Tabloyu Görüntüleme:**

```
bq show PROJECT_ID:DATASET_NAME.TABLE_NAME
```

9. **Bir Tabloyu Silme:**

```
bq rm PROJECT_ID:DATASET_NAME.TABLE_NAME
```

10. **Bir SQL Dosyasını Çalıştırma:**

```
bq query --use_legacy_sql=false --format=prettyjson < QUERY_FILE.sql
```

11. **Çoklu Dosya İçeriğini Veri Setine Aktarma:**

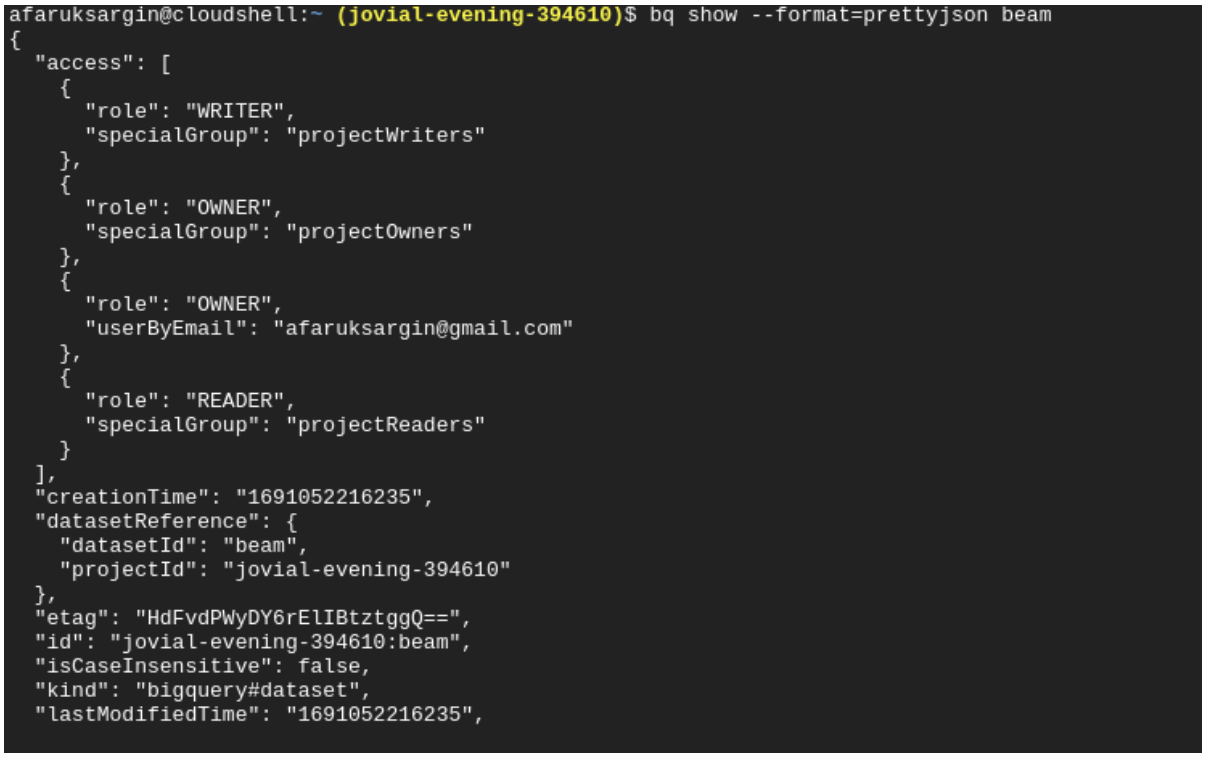
```
bq load --source_format=CSV PROJECT_ID:DATASET_NAME.TABLE_NAME 'gs://BUCKET_NAME/*.csv' SCHEMA
```

12. **Bir Veri Seti Silme:**

```
bq rm -r -f PROJECT_ID:DATASET_NAME
```

13. **Bir Sorgunun Sonuçlarını Dosyaya Aktarma:**

```
bq query --destination_table PROJECT_ID:DATASET_NAME.TEMP_TABLE --use_legacy_sql=false 'SELECT * FRO
M `PROJECT_ID.DATASET_NAME.TABLE_NAME`'
bq extract PROJECT_ID:DATASET_NAME.TEMP_TABLE DESTINATION_FILE
```



14. **Csv Dosyasını Yükleme**

```
bq load --source_format=CSV PROJECT_ID:DATASET_NAME.TABLE_NAME gs://BUCKET_NAME/PATH_TO_CSV_FILE.csv SCHEMA
```

Python Kodu ile Tablo oluşturma Ve Veri Yükleme

```
from google.cloud import bigquery

# JSON hizmet hesabı kimlik bilgileri ile istemciyi oluşturun
client = bigquery.Client.from_service_account_json('keys.json')

# Oluşturmak istediğiniz tablonun bilgilerini tanımlayın
table_id = 'PROJECT_ID.DATASET_NAME.TABLE_NAME'
schema = [
    bigquery.SchemaField('column1', 'STRING', mode='NULLABLE'),
    bigquery.SchemaField('column2', 'INTEGER', mode='NULLABLE'),
    # Diğer sütunlar burada eklenebilir
]

# Tabloyu oluşturun
table = bigquery.Table(table_id, schema=schema)
table = client.create_table(table)  # Tabloyu oluşturun

# Eklenecek verileri hazırlayın
rows_to_insert = [
    ('value1', 123),
    ('value2', 456),
    # Diğer veriler burada eklenebilir
]

# Verileri ekleyin
errors = client.insert_rows(table, rows_to_insert)

if not errors:
    print('Veriler tabloya başarıyla eklendi.')
else:
    print('Veriler eklenirken hata oluştu:', errors)
```

CSV İçerisinden Veri Aktarımı

```
from google.cloud import bigquery

# JSON hizmet hesabı kimlik bilgileri ile istemciyi oluşturun
client = bigquery.Client.from_service_account_json('keys.json')

# Verilerin yükleneceği tablonun bilgilerini tanımlayın
table_id = 'PROJECT_ID.DATASET_NAME.TABLE_NAME'

# CSV dosyasının yolunu belirtin
csv_file_path = 'path/to/your/file.csv'
```

```
# CSV dosyasındaki verileri tabloya eklemek için LoadJobConfig konfigürasyonunu ayarlayın
load_config = bigquery.LoadJobConfig()
load_config.source_format = bigquery.SourceFormat.CSV
load_config.skip_leading_rows = 1  # İlk satırı atla (başlık)
load_config.autodetect = True  # Sütunları otomatik olarak tanımla

# Verileri tabloya yükle
load_job = client.load_table_from_uri(
    csv_file_path, table_id, job_config=load_config
)

load_job.result()  # İşlem tamamlanmasını bekle

print('CSV verileri tabloya başarıyla eklendi.')
```

Cloud Storageden BigQuerye Veri Aktarma

```
from google.cloud import bigquery
from google.cloud import storage

# JSON hizmet hesabı kimlik bilgileri ile BigQuery ve Storage istemcilerini oluşturun
bq_client = bigquery.Client.from_service_account_json('bq_keys.json')
storage_client = storage.Client.from_service_account_json('storage_keys.json')

# Verilerin yükleneceği tablonun bilgilerini tanımlayın
table_id = 'PROJECT_ID.DATASET_NAME.TABLE_NAME'

# Yüklemek istediğiniz CSV dosyasının yolunu belirtin
csv_file_path = 'gs://BUCKET_NAME/PATH_TO_CSV_FILE.csv'

# BigQuery tablosunun mevcut şemasını alın (isteğe bağlı)
table = bq_client.get_table(table_id)
schema = table.schema

# Veriyi BigQuery tablosuna yükleme işlemi için LoadJobConfig konfigürasyonunu ayarlayın
load_config = bigquery.LoadJobConfig()
load_config.source_format = bigquery.SourceFormat.CSV
load_config.skip_leading_rows = 1  # İlk satırı atla (başlık)
load_config.schema = schema  # Tablonun şemasını ayarlayın (isteğe bağlı)

# BigQuery tablosuna verileri yükle
load_job = bq_client.load_table_from_uri(
    csv_file_path, table_id, job_config=load_config
)

load_job.result()  # İşlem tamamlanmasını bekle

print('CSV verileri BigQuery tablosuna başarıyla yüklendi.')
```

Yukarıdaki örnek kodda `bq_keys.json` ve `storage_keys.json` dosyalarını kendi JSON hizmet hesabı kimlik bilgilerinizle güncellemelisiniz. `"PROJECT_ID.DATASET_NAME.TABLE_NAME"` kısmını oluşturduğunuz BigQuery projesinin, veri kümesinin ve tablosunun isimleriyle değiştirmeyi unutmayın. `csv_file_path`

değişkenini, yüklemek istediğiniz CSV dosyasının Google Cloud Storage yoluna göre güncellemelisiniz.

Ayrıca, BigQuery tablosunun mevcut şemasını alarak veriyi uygun sütunlara eşleştirmeniz gerekebilir. Şema bilgisine ihtiyaç duymazsanız, `load_config.schema` satırını kaldırabilirsiniz.