



EBook Gratis

APRENDIZAJE google-bigquery

Free unaffiliated eBook created from
Stack Overflow contributors.

#google-
bigquery

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con google-bigquery.....	2
Observaciones.....	2
Examples.....	2
Instalación o configuración.....	2
Analizando 50 mil millones de páginas vistas de Wikipedia en 5 segundos (tutorial para pri.....	2
Preparar.....	2
Vamos a consultar.....	2
Análisis de costos.....	3
Cargando datos en BigQuery.....	4
Aprende más.....	4
¡Sigue por más!.....	4
Capítulo 2: Integración de BigQuery de Google con la aplicación web.....	6
Examples.....	6
Integración de la API BigQuery de Google con la aplicación web.....	6
Creditos.....	9

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [google-bigquery](#)

It is an unofficial and free google-bigquery ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official google-bigquery.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con google-bigquery

Observaciones

Esta sección proporciona una descripción general de qué es google-bigquery y por qué un desarrollador puede querer usarlo.

También debe mencionar cualquier tema importante dentro de google-bigquery, y vincular a los temas relacionados. Dado que la Documentación para google-bigquery es nueva, es posible que deba crear versiones iniciales de esos temas relacionados.

Examples

Instalación o configuración

Instrucciones detalladas para configurar o instalar google-bigquery.

Analizando 50 mil millones de páginas vistas de Wikipedia en 5 segundos (tutorial para principiantes de BigQuery)

¡Hola a todos! Esta es una demostración que me encanta ejecutar para personas que comienzan con BigQuery. Así que vamos a ejecutar algunas consultas simples para ayudarle a empezar.

Preparar

Necesitará un proyecto de Google Cloud:

1. Vaya a <http://bigquery.cloud.google.com/> .
2. Si le dice que cree un proyecto, siga el enlace para crear un proyecto y cree un proyecto.
3. Vuelve a <http://bigquery.cloud.google.com/> .

Notas:

- No necesitas una tarjeta de crédito. Todos obtienen un 1TB gratis para el análisis cada mes.
- BigQuery cobra por consulta. Antes de ejecutar una consulta, podrá ver cuánto cuesta cada consulta.

Vamos a consultar

1. Encuentre las páginas vistas para mayo de 2015 en https://bigquery.cloud.google.com/table/fh-bigquery:wikipedia.pagecounts_201505

Nota : Google protege sus datos con los más altos estándares de seguridad (PCI, ISO, HIPAA, SOC, etc.), pero también es fácil compartir datos si así lo desea, como lo hice aquí.

2. Esta tabla tiene 5 columnas: el *título de la fecha y hora solicita el contenido de tamaño* . Básicamente dicen "esta página de wikipedia en este idioma tenía tantas solicitudes a esta hora".
3. Esta tabla tiene casi 6 mil millones de filas (379 GB de datos).
4. Para saber cuántas páginas vistas tuvo Wikipedia durante mayo, puede sumar todas las 6 mil millones de líneas de solicitudes:

```
SELECT SUM(requests)
FROM [fh-bigquery:wikipedia.pagecounts_201505]
```

5. ¿Te diste cuenta de lo rápido que era eso? (1.8s transcurridos, 43.1 GB procesados para mí)
6. Hagamos algo más complejo. Ejecutemos una expresión regular sobre estos 6 mil millones de filas. ¿Qué tan rápido podría ser esto?

```
SELECT SUM(requests) req, title
FROM [fh-bigquery:wikipedia.pagecounts_201505]
WHERE REGEXP_MATCH(title, 'Red.*t')
GROUP BY title
ORDER BY req DESC
LIMIT 100
```

7. ¿Qué tan rápido fue para ti? ¿Encontraste Reddit en los resultados?

Análisis de costos

1. Esta última consulta procesó 269 GB: más de un cuarto del terabyte mensual gratuito. ¿Por qué?
2. BigQuery analiza las columnas que procesa en su consulta. 'título' es una gran columna que contiene texto. La columna de 'solicitudes' solo tiene 43.1 GB.
3. Para que su terabyte libre dure, extraiga los datos en tablas más pequeñas. Por ejemplo, tengo una tabla con solo las [65,000 páginas de páginas de Wikipedia en inglés más visitadas](#) . La misma consulta procesa solo 1.18 GB, puede ejecutar casi 1000 de ellos gratis por mes.

```
SELECT SUM(requests) req, title
FROM [fh-bigquery:wikipedia.pagecounts_201408_en_top65k]
WHERE REGEXP_MATCH(title, 'Red.*t')
GROUP BY title
ORDER BY req DESC
LIMIT 100
```

4. No se pueden crear tablas con el terabyte mensual gratuito, solo para análisis. Active sus \$

300 gratuitos para las nuevas cuentas de Google Cloud Platform, o pídamelo que haga un extracto por usted. Estaré feliz de hacerlo.

Cargando datos en BigQuery

Para cargar datos en BigQuery, deberá activar la facturación de su proyecto; pruébelo con sus \$ 300 gratuitos para nuevas cuentas.

0. Cree un conjunto de datos en su proyecto para cargar los datos en:

<https://i.imgur.com/FRCIJ3K.jpg> .

1. Encuentre los registros sin procesar compartidos por Wikipedia en

<https://dumps.wikimedia.org/other/pagecounts-raw/>

2. ingrese uno de estos archivos en su computadora, como

<https://dumps.wikimedia.org/other/pagecounts-raw/2015/2015-06/pagecounts-20150616-160000.gz>

3. Instala la herramienta 'bq'. <https://cloud.google.com/bigquery/bq-command-line-tool>

4. Cargarlo en BigQuery:

```
bq load -F" " --quote "" YourProjectName:DatasetName.pagecounts_20150616_16 pagecounts-20150616-160000.gz language,title,requests:integer,content_size:integer
```

5. Espere un par de minutos. Mientras espera, déjeme explicarle esa línea: este no es un archivo CSV, es un archivo separado por espacios (-F "") que no usa comillas (--quote ""), elegimos una tabla de destino en un conjunto de datos en su proyecto (recuerde crear el conjunto de datos primero), elegimos el archivo para cargar y definimos las 4 columnas que tiene este archivo.

6. Tenga en cuenta que BigQuery felizmente ingerirá archivos .gz, hasta cierto tamaño. Para archivos muy grandes, es mejor descomprimirlos y colocarlos en Google Cloud Storage primero. Eso es lo que hice con los [comentarios de reddit](#) que / u / Stuck_In_The_Matrix compiló. Esos archivos eran grandes, pero BigQuery los ingirió en aproximadamente 2 minutos.

Aprende más

Listo para ejemplos más avanzados? Vea [cómo consultar Reddit](#) y [cómo consultar todos los viajes en taxi de Nueva York](#) .

¡Sigue por más!

- Suscríbese a [/ r / bigquery](#) (y vea la barra lateral para obtener más enlaces).
- Sígueme en <https://twitter.com/felipehoffa>

Lea Empezando con google-bigquery en línea: <https://riptutorial.com/es/google-bigquery/topic/4799/empezando-con-google-bigquery>

Capítulo 2: Integración de BigQuery de Google con la aplicación web.

Examples

Integración de la API BigQuery de Google con la aplicación web.

En este tutorial, explicaré cómo integrar la API de BigQuery de Google con la aplicación web. Mi aplicación web obtendrá los datos utilizando BigQuery y trazará un gráfico utilizando d3.js y Javascript.

Cada proyecto en la consola de desarrolladores de Google tiene un ID de cliente y usted necesita copiar el ID de cliente y ponerlo como una configuración:

```
var gconfig = {
  'client_id': 'ClientID',
  'scope': 'https://www.googleapis.com/auth/bigquery'
};
```

Se puede acceder a la API de BigQuery de la siguiente manera:

```
$.getScript("https://apis.google.com/js/client.js", function(d) {
  function loadGAPI() {
    setTimeout(function() {
      if (!gapi.client) {
        loadGAPI();
      } else {
        loadBigQuery();
      }
    }, 500);
  }

  function loadBigQuery() {
    gapi.client.load('bigquery', 'v2');
    setTimeout(function() {
      if (!gapi.client.bigquery) {
        loadBigQuery();
      } else {
        onClientLoadHandler();
      }
    }, 500);
  }

  loadGAPI();
});
```

También deberás mencionar la consulta de la que vas a recuperar los datos:

```
function runQuery() {
  var request = gapi.client.bigquery.jobs.query({
    'projectId': "bigdatameetup-83116",
```

```

    'timeoutMs': '30000',
    'query': 'SELECT DATE(date ) as date,SUM(INTEGER(orders)) as total_orders FROM
[bigdatameetup-83116:Demo_Backup.orders] GROUP BY date ORDER BY date LIMIT 1000; '
  });
  request.execute(function(response) {
    var bqData = [];

    response.result.rows.forEach(function(d) {
      bqData.push({"date": d3.time.format("%Y-%m-%d").parse(d.f[0].v),
        "total_orders": +d.f[1].v});
    });

    drawLineChart(bqData);
  });
}

```

El resto es la visualización, es decir, la creación de Line Chart usando d3.js:

```

function drawLineChart(bqData) {
  var WIDTH = config.width, HEIGHT = config.height;
  var Y_AXIS_LABEL = "total_orders";
  var X_DATA_PARSE = d3.time.format("%d-%b-%y").parse;
  var Y_DATA_PARSE = vida.number;
  var X_DATA_TICK = d3.time.format("%b-%y");
  var X_AXIS_COLUMN = "date";
  var Y_AXIS_COLUMN = "total_orders";
  var margin = {top: 20, right: 20, bottom: 30, left: 50},
    width = WIDTH - margin.left - margin.right,
    height = HEIGHT - margin.top - margin.bottom;
  var x = d3.time.scale()
    .range([0, width]);
  var y = d3.scale.linear()
    .range([height, 0]);
  var xAxis = d3.svg.axis()
    .scale(x)
    .orient("bottom")
    .tickFormat(X_DATA_TICK);
  var yAxis = d3.svg.axis()
    .scale(y)
    .orient("left")
    .tickFormat(function(d) {
      return d / 1000000 + "M";
    });
  var line = d3.svg.line()
    .interpolate("basis")
    .x(function(d) { return x(d.x_axis); })
    .y(function(d) { return y(d.y_axis); });

  var svg = d3.select("#canvas-svg").append("svg")
    .attr("width", width + margin.left + margin.right)
    .attr("height", height + margin.top + margin.bottom)
    .append("g")
    .attr("transform", "translate(" + margin.left + "," + margin.top + ")");

  bqData.forEach(function(d) {
    d.x_axis = d[X_AXIS_COLUMN];
    d.y_axis = d[Y_AXIS_COLUMN];
  });

  bqData.sort(function(a, b) {

```

```

    return (new Date(a.x_axis)) - (new Date(b.x_axis));
  });

  x.domain(d3.extent(bqData, function(d) { return d.x_axis; }));
  y.domain(d3.extent(bqData, function(d) { return d.y_axis; }));

  svg.append("g")
    .attr("class", "x axis")
    .attr("transform", "translate(0," + height + ")")
    .call(xAxis);

  svg.append("g")
    .attr("class", "y axis")
    .call(yAxis)
    .append("text")
    .attr("transform", "rotate(-90)")
    .attr("y", 6)
    .attr("dy", ".71em")
    .style("text-anchor", "end")
    .text(Y_AXIS_LABEL);

  svg.append("path")
    .datum(bqData)
    .attr("class", "line")
    .attr("d", line);
}

```

En este ejemplo, he elegido 'Cantidad' como eje x y 'Fecha' como eje y del conjunto de datos público:

nyc_taxi_public

Puede encontrar la muestra de trabajo completa en este enlace.

[BigQuery Integration with WebApplication](#)

Lea Integración de BigQuery de Google con la aplicación web. en línea:

<https://riptutorial.com/es/google-bigquery/topic/6139/integracion-de-bigquery-de-google-con-la-aplicacion-web->

Creditos

S. No	Capítulos	Contributors
1	Empezando con google-bigquery	Community , Felipe Hoffa
2	Integración de BigQuery de Google con la aplicación web.	Eduard Malakhov , JL-HN , Sajeetharan