



Introducción a BigQuery y realización de consultas con Java

Requisitos Previos:

- IDE Eclipse
- Cuenta en Google.
- Java Development Kit versión 8 u 11
- Apache Maven 3.0 o posterior

Instrucciones:

BigQuery es un servicio de almacenamiento de datos corporativos en la nube de Google. BigQuery almacena datos mediante un formato de almacenamiento en columnas optimizado para consultas analíticas. BigQuery presenta datos en tablas, filas y columnas, y proporciona compatibilidad total con la semántica de transacción de la base de datos (ACID). El almacenamiento de BigQuery se replica de forma automática en varias ubicaciones para proporcionar una alta disponibilidad.

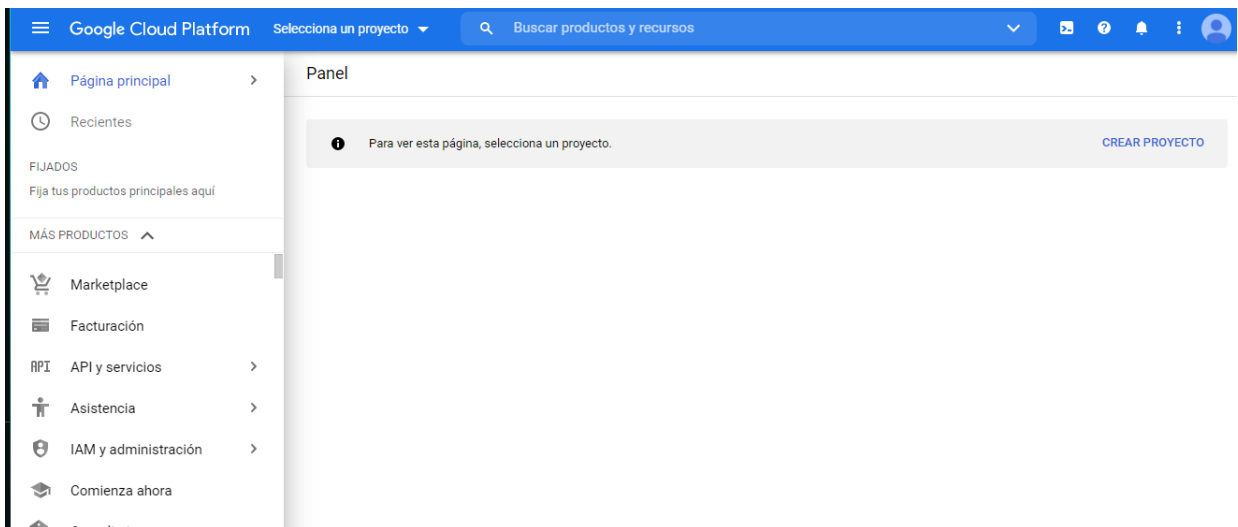
Creación de primer proyecto de BigQuery

Empecemos con la creación de un nuevo proyecto de Google Cloud Platform.

Accedemos a la página console.cloud.google.com e iniciamos sesión con una cuenta de Google.

Es importante que nuestra cuenta de Google tenga activado la prueba gratuita de Google Cloud o en su defecto que nuestro perfil de pago esté completo para evitar problemas de acceso.

Nos presentará una pantalla y daremos click a la opción CREAR PROYECTO.





Ponemos un nombre para nuestro proyecto y en automático nos creará un ID del proyecto el cual no puede ser modificado. Opcionalmente podemos colocar una organización, para este ejemplo lo dejaremos “sin organización”

Google Cloud Platform

Buscar productos y recursos

Proyecto nuevo

⚠ Tienes 12 projects restantes en tu cuota. Solicita un incremento o borra algunos proyectos. [Más información](#)

[MANAGE QUOTAS](#)

Nombre del proyecto *

ID de proyecto: bigquery-337317. No se podrá cambiar más tarde. [EDITAR](#)

Ubicación * [EXPLORAR](#)

Organización o carpeta superior

[CREAR](#) [CANCELAR](#)

Al terminar la creación del proyecto, buscaremos el recurso BigQuery y seleccionamos la primera opción de la sección “Productos y páginas”

Google Cloud Platform

BigQuery

Página principal

Recientes

FIJADOS

Fija tus productos principales aquí

Los marcadores aparecen aquí

MÁS PRODUCTOS

Marketplace

Facturación

API y servicios

Asistencia

IAM y administración

PRODUCTOS Y PÁGINAS

- BigQuery
- Bigtable
- BI Engine BigQuery
- Consultas programadas BigQuery

INSTRUCTIVOS Y DOCUMENTACIÓN

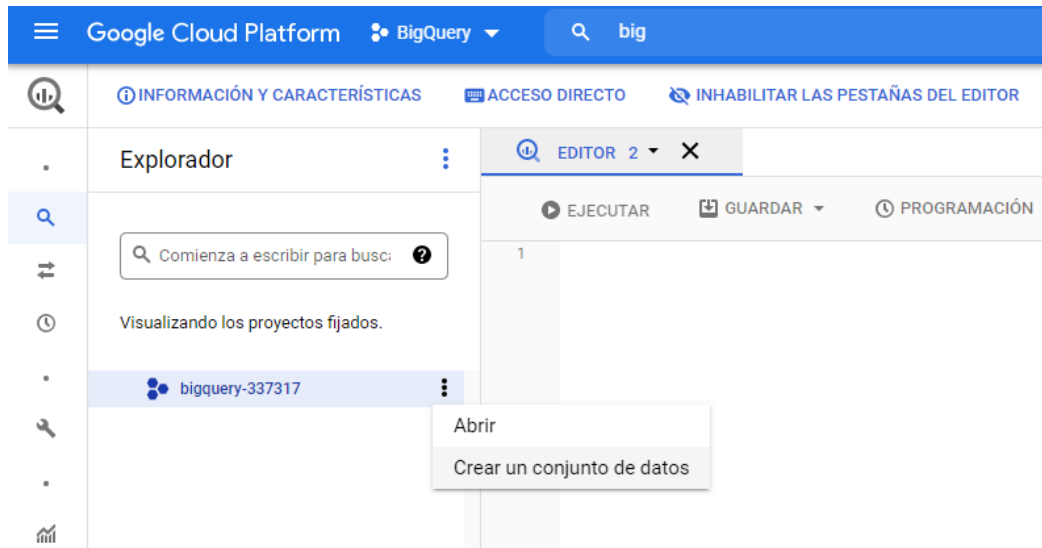
- Importa datos de Cloud Storage a BigQuery Instructivo interactivo
- Analyze Billing data with BigQuery Instructivo interactivo
- Import data from Cloud Storage to BigQuery

<https://console.cloud.google.com/bigquery?referrer=search&authuser=2&project=bigquery-337317&supportedpurview=project>

Veremos una página que nos presentará un explorador donde estarán nuestros proyectos creados y un espacio para realizar consultas directamente.



Crearemos un conjunto de datos que será el equivalente a nuestra base de datos en la nube. Seleccionamos los tres puntos verticales que se encuentran a un costado del nombre de nuestro proyecto y elegimos la opción “Crear un conjunto de datos”



Nos pedirán un nombre cualquiera y elegiremos la región donde estarán alojados los datos. Podemos habilitar un vencimiento si se requiere, así como el administrador de la clave de encriptación.

Crea un conjunto de datos

ID del proyecto
bigquery-337317 [CAMBIO](#)

ID del conjunto de datos *
bigquery
Puede incluir letras, números y guiones bajos

Ubicación de los datos
us (varias regiones en Estados Unidos) [?](#)

Vencimiento predeterminado de la tabla

☐ Habilitar el vencimiento de la tabla [?](#)

Máxima antigüedad predeterminada de la tabla Days

Encriptación

☒ Clave de encriptación administrada por Google
No se requiere configuración

☐ Clave de encriptación administrada por el cliente (CMEK)
Administrar a través de Google Cloud Key Management Service

[CREAR CONJUNTO DE DATOS](#) [CANCELAR](#)

Ahora podemos crear nuestra primera tabla de datos.



Seleccionamos los tres puntos verticales ubicados a un costado del nombre de nuestro conjunto de datos y damos click en la opción “Crear tabla”. Observamos un formulario de creación el cual nos pedirá un origen y un destino.

El origen puede ser desde una tabla en blanco o pueden exportarse desde servicios de almacenamiento e Google como Drive, de terceros como Amazon S3 o desde un archivo local. Para este ejemplo seleccionaremos la opción “Tabla Vacía”

El destino será nuestra nueva tabla alojada en el conjunto de datos creado anteriormente, en el apartado tabla le damos un nombre cualquiera y aremos click en el botón azul “Crear Tabla”

Crear tabla >

Origen

Crear tabla desde
Tabla vacía

Destino

Proyecto *
bigquery-337317 [EXPLORAR](#)

Conjunto de datos *
bigquery

Tabla *
Se permiten letras, signos, números, conectores, guiones o espacios Unicode.

Tipo de tabla
Tabla nativa

Esquema

[CREAR TABLA](#) [CANCELAR](#)

Ahora podemos agregar el esquema con el que se se formará nuestra tabla.

Seleccionamos la opción “Editar esquema” y podemos agregar mediante texto o mediante un formulario los atributos de la tabla.

Editar esquema

Esquema actual

ADD POLICY TAG

Filtro Ingresar el nombre o el valor de la propiedad

<input type="checkbox"/>	Nombre del campo	Tipo	Modo	Etiquetas de políticas	Descripción
No hay resultados para mostrar					

Campos nuevos

☒ Editar como texto

Nombre del campo	Tipo	Modo	Descripción	
<input type="text"/>	STRING	NULLAB...	<input type="text"/>	Longitud máxima

[GUARDAR](#)

[CERRAR](#)



Realización de una consulta con Java

Ahora utilizaremos una base de datos pública que nos provee Google Cloud para realizar una prueba de conexión a BigQuery desde un proyecto Java, así como obtener los resultados de dicha consulta desde el proyecto.

En la sección del explorador de nuestra consola, buscaremos la palabra “public” y Enter. Ampliamos la búsqueda a todos los proyectos y obtendremos el dataset “bigquery-public-data” damos un click en el pin de lado derecho del nombre para fijarlo en nuestro explorador.

Podemos explorar la gran cantidad de conjuntos de datos y tablas que Google tiene para ofrecernos. Para nuestro ejemplo, trabajaremos con el dataset “github-repos” y su tabla “commits”

Explorador

Visualizando los proyectos fijados.

- geo_whos_on_first
- ghcn_d
- ghcn_m
- github_repos
 - commits
 - contents
 - files
 - languages

comits

ESQUEMA DETALLES VISTA PREVIA

Esquema de la tabla

Filtro Ingresar el nombre o el valor de la propiedad

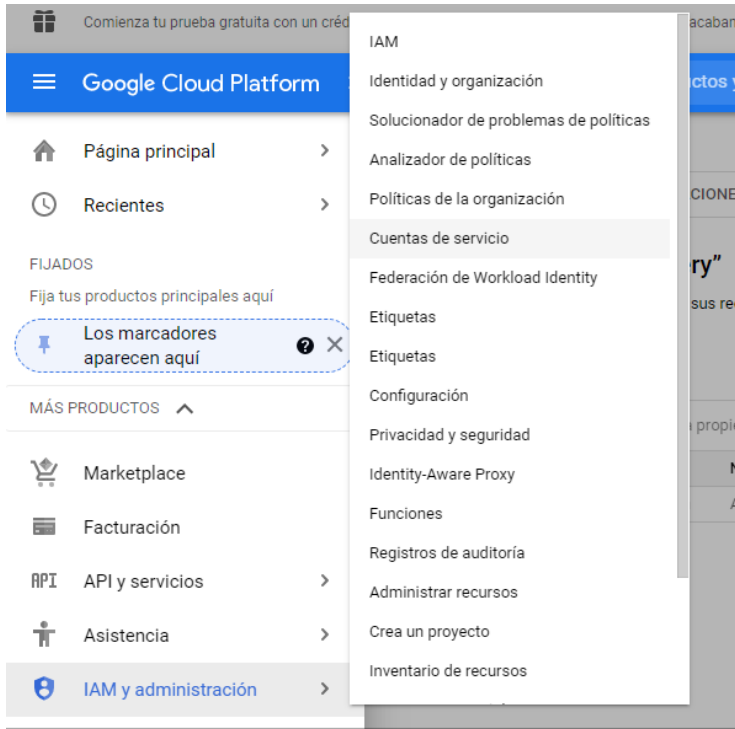
Nombre del campo	Tipo	Modo	Etiquetas de política	Descripción
commit	STRING	NULLABLE		
tree	STRING	NULLABLE		
parent	STRING	REPEATED		
author	RECORD	NULLABLE		
committer	RECORD	NULLABLE		

HISTORIAL PERSONAL HISTORIAL DEL PROYECTO CONSULTAS GUARDADAS

Una parte esencial para nuestro proyecto es la posibilidad de conectarnos a BigQuery fuera de la Google Console Platform.

Para esto, necesitamos una “Cuenta de servicio”, el cual es un documento JSON con las credenciales necesarias para obtener una conexión exitosa.

Para obtener el documento, seleccionamos el menú de tres barras horizontales ubicado al costado izquierdo del logotipo de Google Cloud Platform, nos ubicamos en la opción “IAM y administración” y seleccionamos la opción “Cuentas de Servicio”.



Damos click en el botón “Agregar cuenta de servicio” y nos aparecerá un formulario en donde especificaremos un nombre que deseemos darle a la conexión, así como una descripción opcional.

Para efectos de este ejemplo, saltaremos las secciones opcionales y daremos click en el botón “Listo”.



Para evitar futuros problemas, necesitamos darle permiso a nuestra cuenta para la creación de Jobs en BigQuery. En el menú de la página “IAM Y administración”, seleccionamos la opción “Solucionador de problemas”.

Solucionador de problemas de políticas

Ingresa los siguientes campos para verificar si la llamada a la API te otorgará acceso principal a un recurso.

Si tienes registros de acceso activados, puedes verlos en el [Explorador de registros](#).

Principal (correo electrónico) *

El valor es obligatorio.

Resource permission pairs

Recurso ? Permisos ?

Recomendaciones *

cloudresourcemanager.googleapis.com/projects/bigq bigquery.jobs.create

+ AGREGAR OTRO PAR

COMPROBAR LLAMADA A LA API BORRAR

Escribimos el correo electrónico de la cuenta de Google con la que estamos trabajando, en “Recomendaciones” seleccionamos la única opción que se nos presenta y en permisos escribimos bigquery.jobs.create y damos click al botón “Comprobar Llamada a la API”. Con este proceso, no tendremos problemas de permisos al momento de correr nuestro aplicativo.

Regresamos a la opción “Cuentas de Servicio”. Al final de nuestra cuenta de servicio creada, seleccionaremos el botón del apartado Acciones y seleccionamos la opción “Administrar Claves”.

Cuentas de servicio + CREAR CUENTA DE SERVICIO BORRAR ADMINISTRAR ACCESO

Cuentas de servicio del proyecto “BigQuery”

Una cuenta de servicio representa una identidad de servicio de Google Cloud, como el código en ejecución en las VM de Compute Engine, las apps de A ejecutan fuera de Google. [Obtén más información sobre las cuentas de servicio.](#)

Las políticas de la organización se pueden usar para asegurar las cuentas de servicio y bloquear sus características riesgosas, como el otorgamiento a claves, o la creación misma de cuentas de servicio. [Obtén más información sobre las políticas de la organización para cuentas de servicio.](#)

Filtro Ingresar el nombre o el valor de la propiedad

<input type="checkbox"/>	Correo electrónico	Estado	Nombre ↑	Descripción	ID de clave	Fecha de creac	
<input type="checkbox"/>	javaconnection@bigquery-33731717.iam.gserviceaccount.com	✓	javaConnection	Conectar Java y BigQuery	No hay claves		1064 ⋮

- Administrar detalles
- Administrar permisos
- Administrar claves
- Ver métricas
- Ver registros
- Inhabilitar
- Borrar



En esta nueva página, seleccionamos el menú “Agregar Clave” y la opción “Crear Clave Nueva”

Elegimos la opción JSON, damos click en “Crear” y nos solicitará un lugar de almacenamiento en nuestra computadora, asegúrese de que sea un lugar seguro y guarde el archivo JSON.

Ahora, podemos trabajar sobre nuestro proyecto Java.

En Eclipse IDE o en un IDE con soporte para Java, creamos un nuevo proyecto Java.

Agregaremos las librerías de Google necesarias en nuestro archivo pom.xml, las principales dependencias que utilizaremos son:

```
15 <name>bigquery</name>
16 <description>Demo project for Spring Boot</description>
17 <properties>
18   <java.version>11</java.version>
19 </properties>
20 <dependencyManagement>
21   <dependencies>
22     <dependency>
23       <groupId>com.google.cloud</groupId>
24       <artifactId>libraries-bom</artifactId>
25       <version>24.0.0</version>
26       <type>pom</type>
27       <scope>import</scope>
28     </dependency>
29   </dependencies>
30 </dependencyManagement>
31 <dependencies>
32   <dependency>
33     <groupId>com.google.cloud</groupId>
34     <artifactId>google-cloud-bigquery</artifactId>
35   </dependency>
36 </dependencies>
```

Ahora creamos una clase llamada “SimpleApp”. En algunos IDE, mientras vayamos escribiendo nuestro código se importarán las librerías al momento que queramos solucionar un error en el compilador. En caso de que no se presente esa opción, presentamos las librerías importadas en este proyecto.



```
1 package com.example;
2
3
4 import com.google.auth.oauth2.ServiceAccountCredentials;
5 import com.google.cloud.bigquery.BigQuery;
6 import com.google.cloud.bigquery.BigQueryOptions;
7 import com.google.cloud.bigquery.FieldValueList;
8 import com.google.cloud.bigquery.Job;
9 import com.google.cloud.bigquery.JobId;
0 import com.google.cloud.bigquery.JobInfo;
1 import com.google.cloud.bigquery.QueryJobConfiguration;
2 import com.google.cloud.bigquery.TableResult;
3
4 import java.io.FileInputStream;
5 import java.util.UUID;
```

A continuación, se presenta el código utilizado para consultar desde la base de datos pública, alojada en nuestro proyecto creado anteriormente, los datos sobre repositorios de GitHub que incluyan la etiqueta “BigQuery”.

```
5 import java.util.UUID;
6
7 public class SimpleApp {
8     public static void main(String... args) throws Exception {
9         //Conexión con BigQuery utilizando la clave JSON
10         BigQuery bigquery = BigQueryOptions.newBuilder().setProjectId("bigquery-337317") //Especificar ID del proyecto
11             .setCredentials(
12                 ServiceAccountCredentials.fromStream(new FileInputStream("C:\\Users\\Alan D\\Desktop\\bigquery-337317-a83
13             ))
14             .build().getService();
15         QueryJobConfiguration queryConfig =
16             QueryJobConfiguration.newBuilder(
17                 "SELECT commit, author, repo_name "
18                 + "FROM `bigquery-public-data.github_repos.commits` "
19                 + "WHERE subject like '%bigquery%' "
20                 + "ORDER BY subject DESC LIMIT 10")
21             .setUseLegacySql(false)
22             .build();
23
24         //Creación del Job de consulta
25         JobId jobId = JobId.of(UUID.randomUUID().toString());
26         Job queryJob = bigquery.create(JobInfo.newBuilder(queryConfig).setJobId(jobId).build());
27
28         //Esperar a que termine la consulta
29         queryJob.waitFor();
30
31         //Informar errores
32         if (queryJob == null) {
33             throw new RuntimeException("Job no existe");
34         } else if (queryJob.getStatus().getError() != null) {
35             throw new RuntimeException(queryJob.getStatus().getError().toString());
36         }
37
38         //Obtener los resultados de la consulta
39         TableResult result = queryJob.getQueryResults();
40
41         //Imprimir en consola el resultado de la consulta
42         for (FieldValueList row : result.iterateAll()) {
43             String commit = row.get("commit").getStringValue();
44             FieldValueList author = row.get("author").getRecordValue();
45             String name = author.get("name").getStringValue();
46             String email = author.get("email").getStringValue();
47             String repoName = row.get("repo_name").getRecordValue().get(0).getStringValue();
48             System.out.printf(
49                 "Repo name: %s Author name: %s email: %s commit: %s\n", repoName, name, email, commit);
50         }
51     }
52 }
```

El resultado de la consulta se verá reflejado en la consola de salida de nuestro IDE:



```
Repo name: softkot/gbq Author name: Alexei Volkov email: ee456ea8a89a7f64c1252f65e2ed5b73b7d1a16a@gmail.com commit: 03e93a7fde93486a8dedc89f30b1fe6cba9a10df
Repo name: GoogleCloudPlatform/gcp-service-broker Author name: Colleen Briant email: d250ce8a802a94d138591e7bce3e9155b6e2d62d0@google.com commit: 3c053b1fc853c38cb8d75e14c7b0907ebab61bc9
Repo name: xiaohike/claire Author name: Michael Xia email: f0475e9af188ab67090914f3a68973705c4c8a16@gmail.com commit: 50e98758d48c123377a070991bb9a8b1bb0ebafa
Repo name: nokute78/fluent-bit Author name: Jorge Niedbalski email: 5c2dd944dde9e08881bef0894fe7b22a5c9c4b06@calyptia.com commit: 3cf4207dc0704fa0f4235e8581b9c86b9207bc23
Repo name: googleapis/artman Author name: Alexander Fenster email: 6402b6d12bfe4baae7dad3d018f8cbf6b0e7a044@fenster.name commit: 20839f4c23335969f21f865230a10a5584955243
Repo name: googleapis/artman Author name: Andrea Lin email: 98ae65951b4c98a50f41104ea6290adf3dc4b064@google.com commit: 4a451ad468049daled4e4cf5b98f583333dc7006
Repo name: eviljeff/olympia Author name: Andrew Williamson email: 54e2b4f7b7059d88924ec1826fca3d274bc79e5@mozilla.com commit: 986d3c8c44b8ebee8ccdddb99786d215faa51b7
Repo name: radinformatics/som-tools Author name: vsocx email: 9c1581d433637eb0b0802436a714031cccd16cb0@stanford.edu commit: dc207fccc939c19ed540b20f1f59f090c26e80e5
Repo name: slipp4t/terraform-provider-kubernetes Author name: Radek Sinko email: c0f89f684c2e56811603206eabe5a4cea7f50f3e@gmail.com commit: 213707a65689c1537ff6883b663090b23d351125
Repo name: simonsdave/cloudfeaster-services Author name: simonsdave email: 841f3c50a6f57a2fb07f3e96f89e1b9ff6e48b2a@gmail.com commit: 3890de468a16657a1b8a08724a1e5d80bad6135d
```

Con este programa, fuimos capaces de conectarnos a BigQuery y extraer información almacenada en nuestro proyecto.

Repositorio GitHub:

El proyecto generado en este ejemplo puede ser descargado en el siguiente repositorio de GitHub:

<https://github.com/PADSA-github/Cloud/tree/main/Google-Cloud/Google-BigQuery-Java>