Engineer

BigQuery

el almacén de Google para consultar tablas muy grandes

Qué es

BigQuery es un servicio gestionado de análisis SQL en Google Cloud: nadie tiene que aprovisionar un servidor para utilizarlo, las tablas se almacenan columna por columna y cada consulta se paga por los bytes que lee o con cargo a una capacidad de cómputo reservada de antemano. También se comercializa y se busca como Google BigQuery, y el SQL que utiliza es GoogleSQL, un dialecto que sigue el estándar ANSI y añade funciones propias.

Su diseño se basa en separar el almacenamiento del cómputo. Las tablas residen en la capa de almacenamiento de Google, y cada consulta utiliza capacidad de cómputo mientras se ejecuta. Como cada columna se almacena por separado, una consulta que nombra tres columnas de una tabla de cincuenta lee esas tres y deja las demás en disco. Por eso, la optimización es distinta de la de un servidor PostgreSQL. En lugar de diseñar índices B-tree y dimensionar la memoria, el trabajo se centra en la estructura de la tabla: se puede particionar por una columna de fecha o de marca de tiempo, por el momento de ingesta o por un intervalo de enteros, y agrupar mediante clustering por hasta cuatro columnas para que las filas relacionadas compartan bloques de almacenamiento.

BigQuery pertenece a la familia de los almacenes de datos, y los productos con los que suele compararse son Snowflake y Amazon Redshift, además de la opción de mantener el análisis en la base de datos transaccional que la empresa ya utiliza. Está diseñado para consultas que recorren millones o miles de millones de filas y devuelven unos pocos totales. Las búsquedas y actualizaciones de una sola fila que una aplicación realiza en cada solicitud corresponden a una base de datos como PostgreSQL o SQL Server, y una configuración habitual utiliza ambas, con un flujo de procesamiento que lleva los datos de una a otra.

Utilice el concepto en una decisión

Aplique «¿Qué es BigQuery?» a una decisión real. Podemos ayudarle a relacionar el concepto con el sistema que está construyendo o comprando. Cifras de ventas, operaciones y finanzas reunidas para que dos departamentos no lleven dos respuestas distintas a la misma reunión.

Explore nuestros servicios Póngase en contacto →

Compruebe los supuestos detrás del término

Utilice la explicación y los trabajos enlazados para formular preguntas sobre su proyecto. Podemos examinar los supuestos con usted antes de elegir una herramienta. Flujo de datos geográficos 50 veces más rápido — 50x más rapidez en los flujos de procesamiento de datos geográficos

Flujo de datos geográficos 50 veces más rápido Póngase en contacto →

Por qué importa

La decisión que BigQuery plantea a un equipo es cómo quiere pagar por el análisis. La factura depende de las consultas que se escriben, no del tamaño de una máquina que alguien eligió en su día, y eso tiene dos caras.

Con los precios bajo demanda, cada consulta se cobra por los bytes que procesa. Ese modelo encaja con el trabajo irregular: un equipo que ejecuta unos pocos informes pesados a la semana paga por esos informes y por nada entre ellos, sin un servidor inactivo durante la noche. También pone precio a una consulta descuidada. Un `SELECT *` sobre un año de lecturas de sensores cuesta lo mismo tanto si el analista quería una fila como si las quería todas. Los precios por capacidad sustituyen ese modelo por slots, unidades de cómputo adquiridas mediante reservas, lo que convierte la factura en una cifra conocida de antemano y permite planificar un uso intenso y constante a menor coste. Un proyecto puede cambiar entre ambos modelos, por lo que la primera elección se puede revisar cuando se conozca el uso real.

El almacenamiento tiene sus propias reglas. Una tabla o partición que permanece sin modificaciones durante 90 días consecutivos pasa a la tarifa de largo plazo, un 50 % inferior, sin que nadie tenga que trasladarla. Cada conjunto de datos se factura por el tamaño lógico, sin comprimir, de sus datos o por el tamaño físico, comprimido, y para los datos que se comprimen bien el segundo modelo puede resultar más barato.

Hay tres aspectos que conviene sopesar antes de contratarlo. La ubicación de un conjunto de datos, una sola región o la multirregión US o EU, se fija al crearlo y permanece invariable, de modo que es una decisión de cumplimiento normativo que se toma el primer día. El dialecto, la herramienta de línea de comandos `bq` y las bibliotecas cliente son propios de Google, por lo que llevar las consultas a otro almacén requiere modificarlas; los datos en sí salen con facilidad, ya que las

tablas se exportan a CSV, Avro y Parquet. Y la cuestión de la contratación de personal es menos difícil de lo que parece: los analistas que conocen SQL son productivos en cuestión de días, mientras que la habilidad menos común es interpretar una consulta tanto en bytes como en filas.

¿Cuándo es, entonces, BigQuery la opción adecuada? Cuando los datos llegan en grandes volúmenes a los que principalmente se añaden registros, como eventos, registros de actividad o lecturas de contadores; cuando las preguntas que se les hacen cambian de una semana a otra; y cuando el equipo prefiere no administrar servidores de bases de datos para el análisis. Para unos datos que caben holgadamente en un único servidor PostgreSQL, un almacén añade un segundo sistema, una segunda factura y un flujo de procesamiento entre ambos. Su justificación crece con el volumen y con el número de personas que hacen preguntas.

Cómo se aplica en la práctica

Un conjunto de más de 100 bases de datos, repartidas entre PostgreSQL, Microsoft SQL Server y Google BigQuery, contenía principalmente datos geográficos o de series temporales: coordenadas, lecturas de sensores, registros y métricas en tiempo real. BigQuery se encargaba de los datos de series temporales a gran escala, con sus tablas particionadas por tiempo y agrupadas mediante clustering, aplicando compresión y almacenamiento columnar para reducir los costes de almacenamiento y acelerar las lecturas. Las lecturas sobre datos particionados y agrupados mediante clustering en BigQuery, junto con índices, vistas materializadas y cachés en el conjunto de sistemas, redujeron un 40–60 % los tiempos de respuesta de las consultas GIS y de series temporales.

En un flujo de procesamiento de datos geográficos que se había vuelto lento, primero se analizaron el SQL y los modelos de datos en PostgreSQL, MS SQL y Google Cloud BigQuery, lo que puso de manifiesto una indexación insuficiente, consultas ineficientes y restricciones ausentes. Después, cada motor recibió la herramienta que le correspondía: índices B-tree y GiST en PostgreSQL, índices filtrados en MS SQL y clustering en BigQuery, junto con modelos normalizados, particionamiento y consultas reescritas. El registro de logros de la fundadora indica que el flujo de procesamiento en su conjunto pasó a ser 50 veces más rápido; no especifica el conjunto de datos utilizado para la prueba de rendimiento, el hardware ni el procedimiento de comparación.

Y sobre datos de contadores inteligentes de varias regiones, se escribieron más de 500 informes de análisis en SQL puro para PostgreSQL y BigQuery, con CTE, funciones de ventana, subconsultas y uniones geoespaciales sobre identificadores de contadores, marcas de tiempo, coordenadas y metadatos ambientales. Cada informe se entregó con su SQL anotado, para que otra persona del equipo pudiera volver a ejecutarlo y auditarlo, y con notas sobre problemas conocidos de calidad de los datos, como coordenadas GPS ausentes y valores de contadores corruptos. Los informes se convirtieron en una referencia entre departamentos y contribuyeron a una mejora del 10–15 % en la eficiencia de la planificación operativa.

En los tres casos, BigQuery trabajó junto a PostgreSQL en vez de sustituirlo, y cada motor recibió el trabajo que mejor sabe hacer.

Dónde surgen los problemas

Una consulta con `LIMIT 10` se factura por la tabla completa. En una tabla sin clustering, `LIMIT` recorta el resultado, no los bytes leídos, y `SELECT *` lee todas las columnas existentes. Un vistazo rápido a los datos se convierte en una lectura completa. Indique las columnas necesarias para responder a la pregunta y, para ver una muestra de filas, utilice la vista previa de tablas de la consola o `bq head`, que no tienen coste.

Una tabla particionada se sigue leyendo de principio a fin. Las particiones solo se omiten cuando el filtro sobre la columna de particionamiento es una constante que el motor puede evaluar antes de leer. Comparar la columna con el resultado de una subconsulta, envolverla en una función o realizar operaciones aritméticas sobre ella hace que la consulta recorra todas las particiones. Hay dos medidas de protección útiles. Una tabla puede exigir un filtro de partición, de modo que una consulta sin uno utilizable se detenga con un error en lugar de ejecutarse. Y una ejecución de prueba informa de los bytes que procesaría una consulta sin ejecutarla ni cobrar por ella:

```
bq query --use_legacy_sql=false --dry_run \
  'SELECT meter_id, kwh
   FROM energy.readings
   WHERE reading_date = DATE "2026-10-01"'
```

Se añade clustering y los bytes leídos apenas cambian. El clustering ordena los bloques de almacenamiento por hasta cuatro columnas en secuencia, y el orden importa: una tabla agrupada por fecha del pedido, país y estado no está organizada para un filtro que solo use país y estado. Las tablas de menos de 64 MB obtienen poco beneficio, y en una tabla con clustering los bytes realmente leídos solo se conocen cuando termina la consulta. Ordene las columnas de clustering según los filtros que utilizan las consultas. Cuando haya que limitar el coste antes de ejecutar nada, apóyese en el particionamiento, cuya exclusión de particiones ya se tiene en cuenta en una ejecución de prueba, y establezca un máximo de bytes facturados para que una consulta que supere el límite se detenga antes de generar cargos.

Se rechaza una consulta que une dos conjuntos de datos. Cada conjunto de datos reside en una ubicación y cada consulta se ejecuta en una ubicación, por lo que todos los conjuntos de datos a los que accede deben compartirla. Los equipos que dejan que cada proyecto adopte su propio valor predeterminado acaban con tablas que solo pueden unirse después de copiar una de ellas a la ubicación de la otra. Elija la ubicación una sola vez, teniendo en cuenta el cumplimiento normativo y el origen de los datos, y cree allí todos los conjuntos de datos de forma deliberada.

Lecturas adicionales

- Descripción general de BigQuery — la descripción del servicio que ofrece Google, sus capas separadas de almacenamiento y cómputo y lo que gestiona por usted.
- Introducción a las tablas particionadas — los tipos de particionamiento, su granularidad y sus límites.
- Consultar tablas particionadas — qué filtros excluyen particiones y cuáles hacen que una consulta las recorra todas.
- Introducción a las tablas con clustering — cómo el orden de las columnas determina la exclusión de bloques y qué supone para las estimaciones de costes.
- Estimar y controlar los costes — ejecuciones de prueba, máximo de bytes facturados y formas gratuitas de obtener una vista previa de una tabla.
- Ubicaciones de BigQuery — regiones, multirregiones y reglas para las consultas que acceden a más de un conjunto de datos.

Glosario

Elija según sus restricciones reales

Traiga un conjunto de datos representativo, las preguntas que debe responder y el cuello de botella actual. Podemos acordar cómo medir la calidad, la capacidad de procesamiento y la recuperación.

Vea cuánto cuesta Póngase en contacto →

Haga práctica la terminología

Pregunte cómo se aplica «¿Qué es BigQuery?» a su organización. Basta un ejemplo concreto de su sistema actual para iniciar la conversación.

Explore nuestros servicios

welcome@engineer.company +45 50 10 61 84 Copenhague, Dinamarca

Llévese esta página

Tarjeta de contacto

Creado con esmero y cariño por D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Derechos de autor © 2025 – actualidad · Engineer ApS · Todos los derechos reservados.

Impreso desde https://engineer.company/es/glossary/bigquery/