Engineer

Lago de datos

≋ almacén que conserva datos en bruto en su forma original

Qué es

Un lago de datos es un almacén que conserva los datos en bruto tal como llegan de su fuente, con el máximo nivel de detalle que esta proporciona, para poder responder más adelante a preguntas que todavía nadie ha planteado. En la práctica, los datos se guardan en archivos, a menudo en un formato orientado a columnas como Apache Parquet, dentro de un sistema de archivos distribuido o un almacenamiento de objetos, y los motores de consulta leen esos archivos directamente.

El término procede de James Dixon, de Pentaho. En una entrada de blog fechada el 14 de octubre de 2010, escrita la semana en que Pentaho anunció su primera versión para Hadoop, resumió lo que había oído a las empresas que utilizaban Hadoop. Ocho o nueve de cada diez trabajaban con datos estructurados o semiestructurados, en lugar de no estructurados, y normalmente procedían de una única aplicación o sistema. Se conocían algunas preguntas sobre ellos, surgirían muchas más después, varios grupos de usuarios querían plantearlas y el volumen no cabía en una base de datos relacional, ya fuera por razones técnicas o porque hacerlo no resultaba económicamente razonable. La respuesta habitual había sido elegir los atributos más interesantes y agregarlos en un data mart. Eso solo responde a las preguntas elegidas de antemano, y la agregación oculta para siempre el nivel más fino de detalle.

Dixon propuso en su lugar una imagen. Un data mart es como el agua embotellada: limpia, envasada y con una forma que facilita su consumo. Un lago de datos, en sus palabras, es «una gran masa de agua en un estado más natural». Una fuente lo mantiene lleno, y cualquiera que tenga una pregunta puede acercarse a mirar, entrar en el agua o tomar una muestra.

Cuatro años después, Dixon volvió sobre el término porque se estaba utilizando con muchas definiciones distintas. Reiteró la suya: un lago contiene datos de una sola fuente, sin agregar previamente, y una organización puede tener varios lagos. Los datos de muchos sistemas unidos entre sí eran, según su imagen, un jardín acuático más que un lago. También dejó claro que, en 2010, un lago no sustituía a una base de datos, un data mart ni un almacén de datos. Varias de las definiciones a las que respondía describían un único almacenamiento compartido de datos originales de muchos sistemas. Ese es el sentido más amplio del término, y conviene saber a cuál se refiere un colega.

Utilice el concepto en una decisión

Aplique «¿Qué es un lago de datos?» 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

Cada resumen es una decisión sobre qué preguntas importan. Un total diario de ventas por tienda indica cuánto vendió cada tienda. No puede indicar a qué hora hubo más actividad, qué productos miró la gente sin comprarlos o si un cliente volvió tres veces antes de hacer un pedido. Cuando solo se conserva el total, ya no se pueden responder esas preguntas, porque las filas que permitirían hacerlo se desecharon al calcularlo. Dixon sostenía que las preguntas llegan después de los datos, por lo que los registros originales deben conservarse con el máximo detalle y agregarse una vez conocida la pregunta.

Eso no deja obsoleto al almacén de datos. Dixon describía cómo los data marts y almacenes de datos de las organizaciones se alimentaban con agregaciones tomadas del lago, mientras que las preguntas puntuales se dirigían al propio lago. El almacén de datos sigue siendo el lugar de las cifras que la organización ha acordado. Se acude al lago cuando una pregunta nueva necesita un detalle que el almacén nunca conservó, y para reconstruir una tabla del almacén cuando cambia su definición.

El formato de archivo importa tanto como la idea. Parquet guarda cada columna de un grupo de filas en su propio bloque, y los metadatos al final del archivo registran dónde empieza cada bloque. El lector consulta primero esos metadatos y después recupera solo los bloques de columnas que necesita, de modo que una consulta que utiliza tres columnas de cuarenta no lee las otras treinta y siete. Como Parquet es un formato abierto y cada archivo incorpora sus propios metadatos, cualquier motor que lea Parquet puede consultar la misma carpeta, incluido DuckDB en un portátil, sin un paso de exportación.

Cómo se aplica en la práctica

Una tienda en línea registra cada visualización de producto, cada acción de añadir al carrito y cada compra. En lugar de sumarlas en totales diarios, conserva todos los eventos y los escribe en archivos Parquet, con una carpeta por día. En DuckDB 2.0, ambos pasos son SQL corriente:

```
CREATE TABLE shop_events AS
SELECT * FROM (VALUES
  (TIMESTAMP '2026-10-26 09:14:03', 'view',     'boots-41', NULL::INTEGER),
  (TIMESTAMP '2026-10-26 09:15:40', 'cart',     'boots-41', NULL),
  (TIMESTAMP '2026-10-26 09:17:12', 'purchase', 'boots-41', 1200),
  (TIMESTAMP '2026-10-27 11:02:55', 'view',     'jacket-m', NULL),
  (TIMESTAMP '2026-10-27 11:03:30', 'view',     'boots-41', NULL),
  (TIMESTAMP '2026-10-27 11:05:01', 'cart',     'jacket-m', NULL)
) t(occurred_at, event, product, amount);

COPY (SELECT *, CAST(occurred_at AS DATE) AS day FROM shop_events)
TO 'events' (FORMAT parquet, PARTITION_BY (day));
```

`PARTITION_BY` solo acepta nombres de columna, no expresiones, por lo que el día se calcula en el `SELECT` . DuckDB escribe una carpeta por valor, con el formato clave=valor que su documentación denomina particionamiento Hive:

```
events/day=2026-10-26/data_0.parquet
events/day=2026-10-27/data_0.parquet
```

Una pregunta sobre un día filtra por esa columna:

```
SELECT event, count(*) AS n
FROM   read_parquet('events/*/*.parquet', hive_partitioning = true)
WHERE  day = DATE '2026-10-27'
GROUP  BY event
ORDER  BY n DESC;
-- view 2, cart 1
```

DuckDB traslada un filtro sobre una columna de partición a los nombres de las carpetas y omite los archivos que no pueden coincidir. `EXPLAIN ANALYZE` aplicado a esta consulta indica que se ha examinado uno de los dos archivos. Con un año de carpetas diarias, la misma pregunta abriría una de las 365 carpetas. La columna `day` se recupera como `DATE` sin necesidad de indicarlo, ya que DuckDB detecta de forma predeterminada fechas, marcas de tiempo y enteros grandes en los nombres de las carpetas. También activa por sí mismo el particionamiento Hive cuando las carpetas siguen el patrón clave=valor, de modo que la opción de estas consultas solo hace explícita la intención.

Ahora, una pregunta que nadie había previsto: ¿qué productos se miraron más?

```
SELECT product, count(*) AS views
FROM   read_parquet('events/*/*.parquet', hive_partitioning = true)
WHERE  event = 'view'
GROUP  BY product
ORDER  BY views DESC, product;
-- boots-41 2, jacket-m 1
```

Una tabla de totales diarios de compras nunca podría responder a eso. Los eventos originales lo hacen con una consulta más sobre la misma carpeta, porque se conservó cada evento al llegar.

El servicio de diseño de almacenes de datos de la empresa describe su objetivo como un único lugar donde todas las cifras coincidan, para que dos departamentos dejen de llevar dos respuestas a la misma reunión. Un lago como este se sitúa antes de ese almacén de datos: el almacén contiene las cifras acordadas y los eventos originales siguen disponibles para la siguiente pregunta.

Dónde surgen los problemas

Archivos que nadie sabe explicar. Una carpeta de archivos Parquet sin un registro de su procedencia, de quién se hace responsable de ellos o de lo que significa cada columna es un lago que solo puede utilizar quien lo construyó. Cuando en 2014 sus críticos afirmaron que un lago no tiene metadatos, Dixon respondió que la idea no decía eso, y añadió después que cualquiera que una datos de distintas fuentes o rastree su linaje necesitará metadatos. Mantenga una entrada de catálogo por conjunto de datos con su fuente, responsable, columnas, momento de carga y periodo de conservación, y haga que escribir esa entrada forme parte de la tarea de carga.

Miles de archivos diminutos. Cada archivo Parquet incorpora su propio bloque de metadatos y un lector tiene que leerlo antes de acceder a los datos. Un flujo que escribe un archivo pequeño por minuto deja más de 40.000 archivos en un mes de 30 días, cada uno de los cuales se abre y analiza en cada recorrido completo. Agrupe las escrituras en menos archivos de mayor tamaño y compacte las particiones antiguas según una programación. El formato de tablas Apache Iceberg registra esa compactación como una instantánea `replace` : varios archivos sustituidos por uno que contiene las mismas filas.

Lectores que ven una carga a medias. Una tarea que escribe diez archivos y se detiene después de seis deja seis archivos en la carpeta, y cualquier proceso que enumere su contenido los leerá. Un directorio corriente no tiene una operación de confirmación. Un formato de tablas resuelve este problema. Iceberg registra los archivos de datos individuales en metadatos, en lugar de directorios completos, y confirma un cambio sustituyendo de forma atómica un archivo de metadatos por el siguiente. Los lectores siempre utilizan una instantánea confirmada, las escrituras nunca son visibles parcialmente y los lectores no adquieren bloqueos.

Columnas renombradas que dividen el historial. Los archivos antiguos conservan el nombre anterior de la columna y todas las consultas tienen que conocer ambos. Iceberg selecciona las columnas de los archivos de datos por un identificador de campo, en lugar de por su nombre, de modo que una columna puede renombrarse, añadirse, eliminarse o moverse, incluso dentro de estructuras anidadas, sin reescribir los archivos ya creados.

Un lago para todos los sistemas. Volcar todas las bases de datos de la organización en un solo lugar y cruzar todos sus datos es lo que Dixon llamó un jardín acuático; defendía utilizar varios tipos de almacenamiento en una empresa, cada uno elegido para los datos y las consultas que atiende. Su consejo era utilizar la mejor herramienta para cada tarea: una base de datos transaccional para fines transaccionales y una base de datos analítica para fines analíticos. Empiece un lago con una sola fuente y preguntas reales pendientes de respuesta, y construya las cifras integradas y acordadas en un almacén de datos alimentado por él.

Datos personales conservados para siempre. Un almacenamiento diseñado para conservarlo todo entra en conflicto con las solicitudes de supresión y los límites de conservación. La versión 2 de Iceberg puede eliminar filas individuales escribiendo archivos de eliminación, sin reescribir los archivos de datos, lo que significa que los bytes originales siguen almacenados. Solo pueden desaparecer cuando se reescriben los archivos de datos y caducan las instantáneas anteriores que todavía los enumeran; entonces los archivos pueden eliminarse físicamente. Evite incluir identificadores directos en los eventos originales cuando sea posible, particione los datos de modo que aplicar el periodo de conservación elimine carpetas enteras y programe la compactación y la caducidad de las instantáneas después de las eliminaciones.

Lecturas adicionales

- Pentaho, Hadoop y los lagos de datos — la entrada de James Dixon de octubre de 2010 que introdujo el término y la comparación con el agua embotellada.
- Una nueva mirada a los lagos de datos — el texto de Dixon de 2014 que retoma la idea: una fuente por lago, el jardín acuático y sus respuestas a los críticos.
- Formato de archivo — la organización de Parquet en grupos de filas, bloques de columnas y metadatos de archivo escritos después de los datos.
- Especificación de tablas de Iceberg — la especificación de Apache Iceberg: instantáneas, confirmaciones atómicas, evolución del esquema por identificador de campo, archivos de eliminación y caducidad de las instantáneas.
- Particionamiento Hive — la documentación de DuckDB sobre carpetas clave=valor, traslado de filtros y detección del tipo de las particiones.

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 un lago de datos?» 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/data-lake/