Engineer

Data Lake

≋ ein Speicher für Rohdaten in ihrer ursprünglichen Form

Was es ist

Ein Data Lake ist ein Speicher, der Rohdaten so aufbewahrt, wie sie aus ihrer Quelle eintreffen, mit dem höchsten von der Quelle gelieferten Detailgrad. Damit lassen sich später auch Fragen beantworten, die bisher niemand gestellt hat. In der Praxis liegen die Daten in Dateien, oft in einem spaltenorientierten Format wie Apache Parquet, auf einem verteilten Dateisystem oder in einem Objektspeicher. Abfrage-Engines lesen diese Dateien direkt.

Der Begriff stammt von James Dixon von Pentaho. In einem Blogbeitrag vom 14. Oktober 2010, erschienen in der Woche, in der Pentaho seine erste Hadoop-Version ankündigte, fasste er zusammen, was ihm Unternehmen mit Hadoop berichtet hatten. Acht oder neun von zehn hatten es mit strukturierten oder teilstrukturierten statt mit unstrukturierten Daten zu tun. Meist stammten sie aus einer einzigen Anwendung oder einem einzigen System. Einige Fragen an die Daten waren bekannt, viele weitere würden später entstehen, und mehrere Nutzergruppen wollten sie stellen. Das Volumen passte aus technischen Gründen oder wegen unvertretbarer Kosten nicht in eine relationale Datenbank. Die übliche Antwort war gewesen, die interessantesten Attribute auszuwählen und zu einem Data Mart zu aggregieren. Das beantwortet nur die vorab gewählten Fragen, und die Aggregation verbirgt die feinsten Details endgültig.

Dixon bot stattdessen ein Bild an. Ein Data Mart ist wie abgefülltes Wasser: gereinigt, verpackt und in eine leicht konsumierbare Form gebracht. Ein Data Lake ist in seinen Worten „ein großes Gewässer in einem natürlicheren Zustand“. Eine Quelle füllt ihn ständig auf. Wer eine Frage hat, kann kommen und schauen, hineinwaten oder eine Probe nehmen.

Vier Jahre später griff Dixon den Begriff erneut auf, weil viele unterschiedliche Definitionen kursierten. Er bekräftigte seine eigene: Ein Lake enthält nicht voraggregierte Daten aus einer Quelle, und eine Organisation kann mehrere Lakes haben. Zusammengeführte Daten aus vielen Systemen waren in seinem Bild ein Wassergarten statt eines Sees. Ebenso deutlich sagte er, dass ein Lake 2010 weder eine Datenbank noch einen Data Mart oder ein Data Warehouse ersetzte. Mehrere der Definitionen, auf die er antwortete, beschrieben einen gemeinsamen Speicher für Rohdaten aus vielen Systemen. Das ist der weiter gefasste Sinn des Wortes. Man sollte wissen, welchen ein Kollege meint.

Das Konzept für eine Entscheidung nutzen

Nutzen Sie „Was ist ein Data Lake?“ für eine konkrete Entscheidung. Wir können helfen, das Konzept mit dem System in Verbindung zu bringen, das Sie entwickeln oder einkaufen. Vertriebs-, Betriebs- und Finanzdaten werden zusammengeführt, damit zwei Abteilungen nicht länger mit zwei Antworten in dieselbe Besprechung kommen.

Unsere Leistungen entdecken Kontakt aufnehmen →

Die Annahmen hinter dem Begriff prüfen

Nutzen Sie die Erklärung und die verlinkten Arbeiten, um Fragen für Ihr Projekt zu formulieren. Wir können die Annahmen gemeinsam mit Ihnen prüfen, bevor ein Werkzeug ausgewählt wird.
Geografische Datenpipeline 50-mal schneller — 50x schnellere Geodaten-Pipelines

Geografische Datenpipeline 50-mal schneller Kontakt aufnehmen →

Warum es wichtig ist

Jede Zusammenfassung entscheidet darüber, welche Fragen zählen. Eine tägliche Umsatzsumme pro Geschäft sagt, wie viel jedes Geschäft verkauft hat. Sie sagt nicht, welche Stunde am stärksten war, welche Produkte Menschen ansahen und nicht kauften oder ob ein Kunde vor der Bestellung dreimal zurückkam. Wird nur die Summe aufbewahrt, gehen diese Fragen endgültig verloren: Die Zeilen, die sie beantworten könnten, wurden bei ihrer Berechnung verworfen. Dixons Argument war, dass die Fragen nach den Daten eintreffen. Die Rohdatensätze sollten daher in ihrer feinsten Granularität erhalten bleiben und erst aggregiert werden, wenn die Frage bekannt ist.

Das macht das Warehouse nicht überflüssig. Dixon beschrieb, wie die Data Marts und Warehouses einer Organisation mit Aggregationen aus dem Lake befüllt werden, während Ad-hoc-Fragen direkt an den Lake gehen. Das Warehouse bleibt der Ort für die Zahlen, auf die sich die Organisation geeinigt hat. Zum Lake geht man, wenn eine neue Frage Details braucht, die das Warehouse nie aufbewahrt hat, und wenn eine Warehouse-Tabelle nach einer geänderten Definition neu aufgebaut werden muss.

Das Dateiformat zählt ebenso viel wie die Idee. Parquet speichert jede Spalte einer Zeilengruppe in einem eigenen Block. Die Metadaten am Dateiende halten fest, wo jeder Block beginnt. Ein Leser prüft zuerst diese Metadaten und ruft dann nur die benötigten Spaltenblöcke ab. Eine Abfrage auf drei von vierzig Spalten liest die übrigen siebenunddreißig deshalb nicht. Parquet ist ein offenes Format, und jede Datei trägt ihre eigenen Metadaten. Daher kann jede Engine mit Parquet-Unterstützung denselben Ordner abfragen, auch DuckDB auf einem Laptop, ohne vorherigen Export.

Wie es in der Praxis aussieht

Ein Onlineshop erfasst jeden Produktaufruf, jedes Hinzufügen zum Warenkorb und jeden Kauf. Statt Tagessummen zu bilden, behält er jedes Ereignis und schreibt die Ereignisse als Parquet-Dateien, mit einem Ordner pro Tag. In DuckDB 2.0 sind beide Schritte gewöhnliches SQL:

```
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` akzeptiert nur Spaltennamen, keine Ausdrücke. Deshalb wird der Tag im `SELECT` berechnet. DuckDB schreibt pro Wert einen Ordner. Die Namen folgen der Form Schlüssel=Wert, die seine Dokumentation als Hive-Partitionierung bezeichnet:

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

Eine Frage zu einem einzelnen Tag filtert auf dieser Spalte:

```
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 reicht einen Filter auf einer Partitionsspalte bis zu den Ordnernamen durch und überspringt Dateien, die nicht passen können. `EXPLAIN ANALYZE` meldet bei dieser Abfrage, dass eine von zwei Dateien gelesen wurde. Bei einem Jahr mit Tagesordnern würde dieselbe Frage einen von 365 Ordnern öffnen. Die Spalte `day` kommt ohne ausdrückliche Vorgabe als `DATE` zurück, weil DuckDB Datumswerte, Zeitstempel und große Ganzzahlen in Ordnernamen standardmäßig erkennt. Auch die Hive-Partitionierung aktiviert es selbst, wenn die Ordner dem Muster Schlüssel=Wert folgen. Die Option in diesen Abfragen macht die Absicht also nur sichtbar.

Nun eine nicht vorab geplante Frage: Welche Produkte wurden am häufigsten angesehen?

```
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
```

Eine Tabelle mit täglichen Kaufsummen könnte das nie beantworten. Die Rohereignisse beantworten es mit einer weiteren Abfrage auf demselben Ordner, weil beim Eingang jedes Ereignis erhalten blieb.

Die Leistung zur Konzeption von Data Warehouses des Unternehmens beschreibt ihr Ziel als einen Ort, an dem alle Zahlen übereinstimmen. So bringen zwei Abteilungen nicht länger zwei Antworten in dieselbe Besprechung. Ein Lake wie dieser liegt vor einem solchen Warehouse: Das Warehouse enthält die abgestimmten Zahlen, und die Rohereignisse bleiben für die nächste Frage verfügbar.

Wo es schiefgeht

Dateien, die niemand erklären kann. Einen Ordner mit Parquet-Dateien ohne Aufzeichnung ihrer Herkunft, Verantwortlichen oder Spaltenbedeutungen kann nur sein Ersteller als Lake nutzen. Als Kritiker 2014 einwandten, ein Lake habe keine Metadaten, antwortete Dixon, dass die Idee das nirgends vorgebe. Später ergänzte er, dass jeder, der Daten verschiedener Quellen zusammenführt oder ihre Herkunft nachverfolgt, Metadaten braucht. Halten Sie pro Datenbestand einen Katalogeintrag mit Quelle, Verantwortlichem, Spalten, Ladezeitpunkt und Aufbewahrungsfrist fest. Machen Sie das Schreiben dieses Eintrags zum Bestandteil des Ladejobs.

Tausende winzige Dateien. Jede Parquet-Datei trägt einen eigenen Metadatenblock, den ein Leser vor den Daten lesen muss. Ein Datenstrom, der pro Minute eine kleine Datei schreibt, hinterlässt in einem Monat mit 30 Tagen mehr als 40.000 Dateien. Jede wird bei jedem vollständigen Scan geöffnet und ausgewertet. Bündeln Sie Schreibvorgänge in weniger, größere Dateien und verdichten Sie alte Partitionen nach Zeitplan. Das Tabellenformat Apache Iceberg hält eine solche Verdichtung als `replace` -Snapshot fest: Mehrere Dateien werden gegen eine mit denselben Zeilen ausgetauscht.

Leser sehen unvollständig geladene Daten. Ein Job, der zehn Dateien schreiben soll und nach sechs stoppt, hinterlässt sechs Dateien im Ordner. Alles, was den Ordner auflistet, liest sie. Ein einfaches Verzeichnis kennt keinen Commit. Ein Tabellenformat behebt das. Iceberg verfolgt einzelne Datendateien über Metadaten statt über ganze Verzeichnisse und schreibt eine Änderung fest, indem es atomar eine Metadatendatei gegen die nächste austauscht. Leser verwenden immer einen festgeschriebenen Snapshot. Schreibvorgänge werden nie teilweise sichtbar, und Leser setzen keine Sperren.

Umbenannte Spalten teilen die Historie auf. Alte Dateien behalten den alten Spaltennamen, und jede Abfrage muss beide kennen. Iceberg wählt Spalten in Datendateien anhand einer Feld-ID statt anhand ihres Namens aus. Eine Spalte kann deshalb umbenannt, hinzugefügt, gelöscht oder verschoben werden, auch innerhalb verschachtelter Strukturen, ohne bereits geschriebene Dateien umzuschreiben.

Ein einziger Lake für alle Systeme. Alle Datenbanken einer Organisation an einen Ort zu schütten und quer über sie zu verknüpfen, nannte Dixon einen Wassergarten. Er sprach sich für mehrere Speicherarten im Unternehmen aus,

jeweils passend zu ihren Daten und Abfragen. Sein Rat war, für jede Aufgabe das beste Werkzeug zu verwenden: eine transaktionale Datenbank für transaktionale Zwecke und eine analytische für analytische. Beginnen Sie einen Lake mit einer Quelle und echten offenen Fragen. Bauen Sie die integrierten, abgestimmten Zahlen in einem daraus befüllten Warehouse auf.

Personenbezogene Daten bleiben für immer. Ein auf vollständiges Aufbewahren ausgelegter Speicher kollidiert mit Löschanträgen und Aufbewahrungsgrenzen. Iceberg Version 2 kann einzelne Zeilen löschen, indem es Löschdateien schreibt, ohne die Datendateien umzuschreiben. Die ursprünglichen Bytes bleiben deshalb im Speicher. Sie können erst verschwinden, wenn die Datendateien neu geschrieben wurden und die älteren Snapshots, die sie noch aufführen, abgelaufen sind. Danach können die Dateien physisch gelöscht werden. Halten Sie direkte Identifikatoren möglichst aus Rohereignissen heraus. Partitionieren Sie so, dass bei Ablauf der Aufbewahrungsfrist ganze Ordner entfallen, und planen Sie nach Löschungen Verdichtung und Snapshot-Ablauf ein.

Weiterführende Lektüre

- Pentaho, Hadoop und Data Lakes — James Dixons Beitrag vom Oktober 2010, der den Begriff und den Vergleich mit abgefülltem Wasser einführte.
- Data Lakes erneut betrachtet — Dixons Nachtrag von 2014: eine Quelle pro Lake, der Wassergarten und seine Antworten auf die Kritiker.
- Dateiformat — der Parquet-Aufbau mit Zeilengruppen, Spaltenblöcken und nach den Daten geschriebenen Dateimetadaten.
- Iceberg-Tabellenspezifikation — die Apache-Iceberg-Spezifikation: Snapshots, atomare Commits, Schemaänderungen über Feld-IDs, Löschdateien und Snapshot-Ablauf.
- Hive-Partitionierung — DuckDBs Dokumentation von Schlüssel=Wert-Ordnern, Filterweitergabe und Erkennung von Partitionstypen.

Glossar

Für Ihre tatsächlichen Rahmenbedingungen auswählen

Bringen Sie einen repräsentativen Datensatz, die Fragen, die er beantworten muss, und den aktuellen Engpass mit. Wir können vereinbaren, wie Qualität, Durchsatz und Wiederherstellung gemessen werden.

Preise ansehen Kontakt aufnehmen →

Die Begriffe praktisch nutzbar machen

Fragen Sie, wie „Was ist ein Data Lake?“ auf Ihre Organisation zutrifft. Ein konkretes Beispiel aus Ihrem bestehenden System reicht aus, um das Gespräch zu beginnen.

Unsere Leistungen entdecken

welcome@engineer.company +45 50 10 61 84 Kopenhagen, Dänemark

Nehmen Sie diese Seite mit

Kontaktkarte

Mit Sorgfalt und Liebe gestaltet von D.

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

Copyright © 2025 – heute · Engineer ApS · Alle Rechte vorbehalten.

Gedruckt von https://engineer.company/de/glossary/data-lake/