Engineer

Lac de données

≋ stockage de données brutes dans leur forme d'origine

Ce que c’est

Un lac de données est un espace de stockage qui conserve les données brutes telles qu’elles arrivent de leur source, au niveau de détail le plus fin qu’elle fournit, afin de pouvoir répondre plus tard à des questions encore jamais posées. En pratique, les données sont dans des fichiers, souvent dans un format en colonnes comme Apache Parquet, sur un système de fichiers distribué ou un stockage d’objets, et les moteurs de requêtes lisent directement ces fichiers.

Le terme vient de James Dixon, de Pentaho. Dans un billet daté du 14 octobre 2010, écrit la semaine où Pentaho annonçait sa première version pour Hadoop, il résumait les retours d’entreprises utilisant Hadoop. Huit ou neuf sur dix traitaient des données structurées ou semi-structurées plutôt que non structurées, généralement issues d’une seule application ou d’un seul système. Quelques questions étaient connues, beaucoup d’autres viendraient, plusieurs groupes d’utilisateurs voulaient les poser, et le volume ne pouvait pas être stocké dans une base relationnelle, soit pour des raisons techniques, soit parce que le coût aurait été excessif. La réponse habituelle consistait à choisir les attributs les plus intéressants et à les agréger dans un magasin de données. Cela ne répond qu’aux questions choisies d’avance, et l’agrégation masque définitivement le niveau de détail le plus fin.

Dixon proposait plutôt une image. Un magasin de données est comme de l’eau en bouteille : nettoyée, conditionnée et mise dans une forme facile à consommer. Un lac de données est, selon ses mots, « une grande étendue d’eau dans un état plus naturel ». Une source l’alimente, et toute personne ayant une question peut venir regarder, s’y avancer dans l’eau ou prélever un échantillon.

Quatre ans plus tard, Dixon est revenu sur le terme, utilisé avec de nombreuses définitions. Il a réaffirmé la sienne : un lac contient les données d'une seule source, sans agrégation préalable, et une organisation peut avoir plusieurs lacs. Des données de nombreux systèmes réunies formaient, dans son image, un jardin aquatique plutôt qu'un lac. Il précisait aussi qu'en 2010 un lac ne remplaçait ni une base, ni un magasin, ni un entrepôt de données. Plusieurs définitions auxquelles il répondait décrivaient un stockage partagé unique de données brutes issues de nombreux systèmes. C'est le sens plus large du mot ; mieux vaut savoir lequel emploie votre collègue.

Utilisez le concept pour prendre une décision

Utilisez « Qu'est-ce qu'un lac de données ? » pour éclairer une décision réelle. Nous pouvons vous aider à relier ce concept au système que vous construisez ou achetez. Réunir les chiffres des ventes, des opérations et des finances pour que deux services ne présentent plus deux réponses à la même réunion.

Découvrir nos prestations Contactez-nous →

Vérifiez les hypothèses derrière le terme

Appuyez-vous sur l'explication et les réalisations liées pour formuler les questions de votre projet. Nous pouvons examiner les hypothèses avec vous avant le choix d'un outil. Pipeline géographique accéléré de 50 fois — 50x d'accélération des pipelines de données géographiques

Pipeline géographique accéléré de 50 fois Contactez-nous →

Pourquoi c'est important

Toute synthèse repose sur un choix des questions jugées importantes. Un total quotidien des ventes par boutique dit combien chacune a vendu. Il ne dit pas quelle heure était la plus chargée, quels produits les gens ont regardés sans les acheter, ni si un client est revenu trois fois avant de commander. Dès lors que seul le total est gardé, ces questions sont définitivement perdues : les lignes capables d'y répondre ont été jetées lors du calcul. L'argument de Dixon était que les questions arrivent après les données. Il faut donc garder les enregistrements bruts à leur granularité la plus fine et les agréger une fois la question connue.

Cela ne rend pas l'entrepôt obsolète. Dixon décrivait les magasins et entrepôts de données de l'organisation alimentés par des agrégations tirées du lac, tandis que les questions ponctuelles s'adressent au lac lui-même. L'entrepôt reste le lieu des chiffres sur lesquels l'organisation s'est accordée. Le lac sert lorsqu'une question nouvelle exige un détail que l'entrepôt n'a jamais conservé, et pour reconstruire une table de l'entrepôt quand sa définition change.

Le format des fichiers compte autant que l'idée. Parquet stocke chaque colonne d'un groupe de lignes dans son propre bloc, et les métadonnées en fin de fichier indiquent où commence chacun. Un lecteur consulte d'abord ces métadonnées, puis ne récupère que les blocs de colonnes nécessaires. Une requête portant sur trois colonnes sur quarante ne lit donc pas les trente-sept autres. Parquet étant un format ouvert et chaque fichier portant ses métadonnées, tout moteur capable de le lire peut interroger le même dossier, y compris DuckDB sur un portable, sans étape d'export.

À quoi cela ressemble en pratique

Une boutique en ligne enregistre chaque consultation de produit, ajout au panier et achat. Au lieu d'en faire des totaux quotidiens, elle conserve chaque événement et écrit ces événements dans des fichiers Parquet, un dossier par jour. Dans DuckDB 2.0, les deux étapes s'écrivent en SQL ordinaire :

```
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` n'accepte que des noms de colonnes, pas des expressions ; le jour est donc calculé dans le `SELECT`. DuckDB écrit un dossier par valeur, avec la forme key=value (clé=valeur) que sa documentation appelle partitionnement Hive :

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

Pour répondre à une question sur une journée, on filtre sur cette colonne :

```
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 fait descendre un filtre de colonne de partition jusqu'aux noms de dossiers et ignore les fichiers qui ne peuvent correspondre. Sur cette requête, `EXPLAIN ANALYZE` indique qu'un seul des deux fichiers a été parcouru. Avec une année de dossiers quotidiens, la même question n'ouvrirait qu'un dossier sur 365. La colonne `day` est renvoyée avec le type `DATE` sans qu'on le demande, car DuckDB détecte automatiquement dates, horodatages et grands entiers dans les noms de dossiers, sauf configuration contraire. Il active aussi le partitionnement Hive de lui-même lorsque les dossiers suivent le motif key=value ; l'option des requêtes ne fait donc que rendre l'intention visible.

Voici maintenant une question que personne n'avait prévue : quels produits ont été le plus regardés ?

```
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
```

Une table de totaux quotidiens d'achats ne pourrait jamais y répondre. Les événements bruts le font avec une requête supplémentaire sur le même dossier, parce que chaque événement a été conservé à son arrivée.

La prestation de conception d'entrepôts de données de l'entreprise vise un endroit où tous les chiffres concordent, pour que deux départements cessent d'apporter deux réponses à la même réunion. Un lac comme celui-ci se place en amont d'un tel entrepôt : l'entrepôt garde les chiffres convenus, et les événements bruts restent disponibles pour la question suivante.

Où les problèmes surviennent

Personne ne peut expliquer les fichiers. Un dossier Parquet sans trace de sa provenance, de son propriétaire ou du sens de chaque colonne est un lac dont seule la personne qui l'a créé peut se servir. Quand des critiques affirmaient en 2014 qu'un lac n'avait pas de métadonnées, Dixon répondait que rien dans l'idée ne le disait. Il ajoutait ensuite que relier des sources différentes ou retracer la filiation des données exige des métadonnées. Gardez une entrée de catalogue par jeu de données avec source, propriétaire, colonnes, horodatage du chargement et durée de conservation, et intégrez son écriture à la tâche de chargement.

Des milliers de minuscules fichiers. Chaque fichier Parquet porte son bloc de métadonnées, qu'un lecteur doit lire avant toute donnée. Un flux écrivant un petit fichier par minute laisse plus de 40 000 fichiers pour un mois de 30 jours, chacun ouvert et analysé à chaque parcours complet. Regroupez les écritures en fichiers plus grands et moins nombreux, et compactez les anciennes partitions selon un calendrier. Le format de table Apache Iceberg enregistre ce compactage comme un instantané `replace` : plusieurs fichiers sont remplacés par un seul contenant les mêmes lignes.

Les lecteurs voient un chargement partiel. Une tâche qui doit écrire dix fichiers et s'arrête après six en laisse six dans le dossier, et toute opération qui en énumère le contenu les lira. Un simple répertoire n'a pas de notion de commit. Un format de table corrige cela. Iceberg suit les fichiers de données individuels dans les métadonnées plutôt que les répertoires entiers, puis valide un changement en remplaçant atomiquement un fichier de métadonnées par le suivant. Les lecteurs utilisent toujours un instantané validé, les écritures ne sont jamais partiellement visibles et les lecteurs ne prennent aucun verrou.

Un renommage de colonne coupe l'historique en deux. Les anciens fichiers gardent le nom précédent, et chaque requête doit connaître les deux. Iceberg sélectionne les colonnes des fichiers par identifiant de champ plutôt que par nom. On peut ainsi renommer, ajouter, supprimer ou déplacer une colonne, même dans des structures imbriquées, sans réécrire les fichiers déjà produits.

Un lac pour tous les systèmes. Verser toutes les bases de l'organisation dans un même lieu et les joindre toutes correspond au jardin aquatique de Dixon. Il préconisait plusieurs types de stockage dans l'entreprise, chacun choisi pour ses données et ses requêtes. Son conseil était d'employer le meilleur outil pour

chaque travail : une base transactionnelle pour les transactions, une base analytique pour l'analyse. Commencez un lac par une source avec de vraies questions en attente, et produisez, dans un entrepôt alimenté par ce lac, les chiffres consolidés sur lesquels l'organisation s'accorde.

Les données personnelles restent à jamais. Un stockage conçu pour tout garder se heurte aux demandes d'effacement et aux limites de conservation. Iceberg version 2 peut supprimer des lignes individuelles en écrivant des fichiers de suppression sans réécrire les fichiers de données : les octets d'origine restent donc stockés. Pour pouvoir les éliminer, il faut réécrire les fichiers de données et faire expirer les anciens instantanés qui les répertorient encore ; les fichiers peuvent ensuite être supprimés physiquement. Écartez les identifiants directs des événements bruts quand vous le pouvez, partitionnez de façon à pouvoir supprimer des dossiers entiers à l'expiration de leur durée de conservation, et planifiez compactage et expiration des instantanés après les suppressions.

Pour aller plus loin

- Pentaho, Hadoop et les lacs de données — le billet d'octobre 2010 où James Dixon introduit le terme et la comparaison avec l'eau en bouteille.
- Retour sur les lacs de données — la suite de Dixon en 2014 : une source par lac, le jardin aquatique et ses réponses aux critiques.
- Format des fichiers — groupes de lignes, blocs de colonnes et métadonnées écrites après les données dans Parquet.
- Spécification des tables Iceberg — la spécification Apache Iceberg : instantanés, commits atomiques, évolution du schéma par identifiant de champ, fichiers de suppression et expiration des instantanés.
- Partitionnement Hive — la documentation DuckDB sur les dossiers key=value, la descente des filtres et la détection des types de partitions.

Glossaire

Choisissez selon vos contraintes réelles

Présentez un jeu de données représentatif, les questions auxquelles il doit répondre et le goulot d'étranglement actuel. Nous pouvons convenir de la manière de mesurer la qualité, le débit et la reprise.

Consulter les tarifs Contactez-nous →

Rendez la terminologie concrète

Demandez comment « Qu'est-ce qu'un lac de données ? » s'applique à votre organisation. Un exemple concret tiré de votre système actuel suffit pour engager la discussion.

Découvrir nos prestations

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

Emportez cette page avec vous

Fiche de contact

Créé avec soin et amour par D.

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

Droit d'auteur © 2025 – aujourd'hui · Engineer ApS · Tous droits réservés.

Imprimé depuis https://engineer.company/fr/glossary/data-lake/