1. **Core data concepts**
2. **Classify data**

* Structured data:

+ tất cả dữ liệu đều có chung các trường và các thuộc tính.

+ lược đồ cho các thực thể dữ liệu có cấu trúc là tabular

+ được lưu trong cơ sở dữ liệu, nhiều bảng có thể reference nhau bằng cách sử dụng giá trị khóa trong mô hình quan hệ.

* Semi-structured data:

+ cũng là những thông tin có cấu trúc, nhưng cho phép biến thể trong những cấu trúc đó giữa các entity instances. Ví dụ: phần lớn khách hang có một địa chỉ email, nhưng vẫn có người có nhiều địa chỉ email, và có người lại không có địa chỉ email.

+ một format phổ biến cho semi-structured là JSON

* Unstructed data

+ có những dạng dữ liệu không phải là structured cũng như semi-structured. Ví dụ: dữ liệu, hình ảnh audio, video, binary files

* Data stores: có 2 dạng lưu trữ data thường dùng:

+ file storages

+ databases

1. **File storage**

* Là phần quan trọng trong các hệ điều hành.
* Files có thể lưu trữ trong các hệ thống file trên đĩa cứng của máy tính, hojcw trên các thiết bị có thể di chuyển được như USB, nhưng phần lớn các tổ chức lưu trứ những dữ liệu file quan trong trong tập trung được gọi là shared file storage system, và các hệ thống đó được quản lý bởi cloud, tiết kiệm chi phí, bảo mật, và đáng tin cậy cho dữ liệu.
* **Delimited text files:**

+ dữ liệu được lưu trữ trong format thuần túy với các trường được phân cách và các hang có ký tự kết thúc hàng.

+ phần lớn dữ liệu được lưu trữ trong format CSV: mỗi trường được ngăn cách nhau bởi dấu phẩy và các hang được kết thúc bằng return/newline.

+ có tab-separate values (TSV) và space-delimited

+ phù hợp với structured data được truy cập bởi nhiều úng dụng và dịch vụ với format người human-readable.

**A screenshot of a computer

Description automatically generated with medium confidence**

* **JSON:**

+ hierarchy document schema

+ entities có nhiều thuộc tính

+ mỗi thuộc tính có thể là một object{} (hoặc là tập hợp các objects[{}])

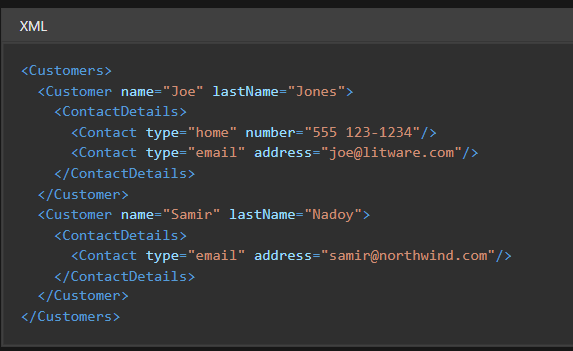
+ phù hợp với cả structed và un-structed.

* **XML:**

**+** humand-readable data format

+ Phần lớn được chuyển sang JSON format để ít phức tạp hơn

+ Sử dụng tag tương tự như html



* **Binary Large Object (BLOB):**

**+** lưu trữ dữ liệu nhị phân,

**+** để theo format dễ đọc thì được map sang các ký tự (ASCII hoặc Unicode)

**+** một số là dịnh dạng tệp, tuy nhiên cụ thể là dữ liệu ko có cấu trúc (images, video, audio, application-specific document)

* **Optimaized file formats:** một số format dữ liệu tối ưu cho storage (nén, chỉ mục,..)

**+** Avro: row-based format: được tạo bởi Apache. Mỗi hàng có header miêu tả structure dữ liệu trong record, header được lưu dưới dạng JSON. Dữ liệu được lưu dạng nhị phân. Sử dụng thông tin trong header để parse dữ liệu nhị phân. Phù hợp cho nén dữ liệu, giảm bộ nhớ lưu trữ và yêu cầu bang thông của mạng

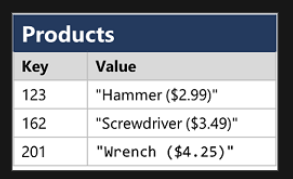
**+ ORC:** tổ chức dưới dạng cột. tối ưu cho các tác vụ đoc và ghi. Chứa các sọc dữ liệu. mỗi stripes chứa dữ liệu của một hoặc 1 tập hợp các cột. mỗi stripes chứa chỉ mục đến hàng trong stripe. Dưới mỗi strips sẽ có những thông itn thống kê (count, sum, mã, min,…)

**+ Parquet:** một dạng lưu dữ liệu dạng cột khác. Dữ liệu mỗi cột được lưu cùng nhau cho mỗi row group. Bao gồm metadata. Đặc biệt được sử dụng trong việc lưu trữ và xử lý các kiêu dữ liệu lồng nhau. Hỗ trợ tốt lược đồ nén và mã hóa

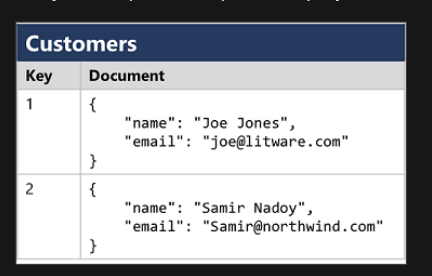
1. **Databases**

* **Relational databases**
* **Non-relational databases (nosql database): có 4 dạn non-relational databases:**

**+** key-valuew database: mỗi record có unique key, và các giá trị có format tùy ý



**+** Document database: một dạng đặc biết của key-value, giá trị được lưu dưới dạng JSON (tối ưu hóa cho parse và query)



+ column family database

A screenshot of a computer

Description automatically generated with medium confidence

+ Grap database (vd: neo4j)

A picture containing font, screenshot, line, circle

Description automatically generated

1. **Transactional data processing**

* Đóng gói các sự kiện cần được theo dõi (ví dụ: chuyển tiền ngân hàng)
* Small, discrete, và là một đơn vị công việc (all or nothing)
* Hệ thống giao tác thường high-volumn, xử lý hàng triệu giao tác trong một ngày
* Dữ liệu phải được truy xuất nhanh
* Online Transactional Processing (OLTP): giải pháp dựa trên hệ cơ sở dữ liệu, tối ưu cho đọc và ghi, để hỗ trợ tải lượng giao tác (created, retrieved, updated, deleted). Đảm bảo tính nhất quán trong dữ liệu được lưu trữ trong csdl. OLTP buộc các giao tác ACID:

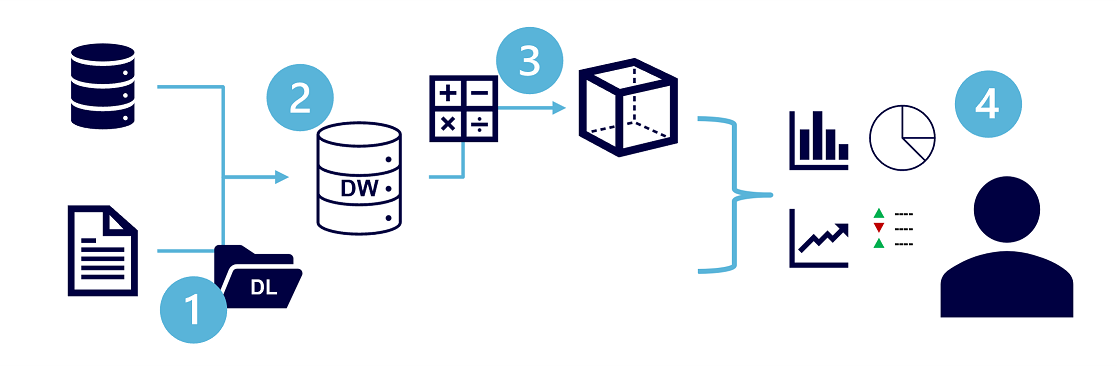
+ Atomic: single unit, all or nothing

+ Consistency: chỉ được lấy dữ liệu trong csdl từ trạng thái valid này sang trạng thái valid khác

+ Isolation: các giao tác đồng thời không bị chồng đè lên nhau và kết quả trong một csdl nhất quán

+ Durability: khi một giao tác đã commit thì dữ liệu được ghi xuống csdl, và ko bị thay đổi hay abort

1. **Analytical data processing**



1. File dữ liệu được lưu trong data lake để phân tích
2. Extract, transform, load (ETL) process copies dât từ files và OLTP database vao trong data warehouse, được tối ưu hóa để đọc. Lược đồ data warehouse dựa trên bảng thự tế chứa numeric values cần được phân tích (vd: sales amounts) với những bảng liên quan dimension chứa những thực hể dựa trên đó có thể phân tích (customer, product)
3. Dữ liệu trong data warehouse có thể được tính gộp và được nạp vào mô hình online analytical processing (OLAP), hoặc cube. Từ bảng thực tế có thể tính intersection của dimensions từ dimension table (vd: tính doanh thu dựa theo ngày, khách hàng, và sản phẩm)
4. Dữ liệu trong datalake, datawarehouse, và trong analytic model có thể được truy vấn để tạo báo cáo, hiển thị hóa, và dashboard.

* Datalakes thường là trong kịch bản xử lý phân tích dữ liệu quy mô lớn, nơi mà khối lượng lớn dữ liệu lưu trữ dựa trên file được thu tập và phân tích
* Data warehouse lữu dữ liệu trong các lược đồ quan hệ để tối ưu hóa cho việc đọc, và truy vấn để hỗ trợ báo váo và hiển thị hóa dữ liệu. datawarehouse có thể yêu cầu giảm chuẩn hóa về data trong OLTP (duplication để làm cho việc truy vấn trở nên nhanh chóng hơn)
* Mô hình OLAP là một nơi lưu trữ kiểu dữ liệu gộp, tối ưu hóa cho việc phân tích. Có thể nhìn dữ liệu gộp tại các cấp dộ khách nhau (vd: doanh số theo region, city, individual address)
* Data scientists có thể làm việc tực tiếp trên file dữ liệu trong data lake
* Data analysts truy vấn bảng trực tiếp từ data warehouse để sinh ra những báo cáo phức tạp và mô phỏng hóa.
* Người sử dụng kinh doanh có thể sử dụng dữ liệu pre-aggregated trong analytic model trong dạng của báo cáo và dashboard

1. **Job roles in the world of data**

* **Database Administrator:**

**+** thiết kế, hiện thực, duy trì, và các khía cạnh vận hành trên csdl (on-premise, cloud).

**+** làm việc với các bên liên quan để hiện thực policies, tools và những quy trình để backup và phục hồi khi có thiên tai hoặc do lỗi từ con người.

**+** quản lý tính an toàn của dữ liệu, và cấp đặc quyền thóng qua dữ liệu, cấp hoặc thu hồi quyền truy cập của người dùng

* **Data Engineer:**

+ phối hợp với các bên liên quan để thiết kế và hiện thực liên hệ dữ liệu. bao gồm: pipelines, các hoạt động làm sạch và chuyển đổi dữ liệu, và dữ liệu lưu trữ cho các hoạt động phân tích

+ sử dụng các công nghệ platform: relational, non-relational db, file stores, và data streams

+ bảo đảm tính bảo mật của dữ liệu và duy trì trong cloud, mở rộng từ on-premise đến cloud.

* **Data Analyst:**

+ khám phá dữ liệu để tìm ra xu hướng và mối liên hệ, thiết kế và xây dựng mô hình phân tích, cho phép phân tích bậc cao thông qua báo cáo và hiển thị hóa dữ liệu

1. **Azure SQL: tên cho các giải pháp csdl quan hệ dựa trên MS sql server bao gồm:**

* Azure SQL DB: PaaS database
* Azure SQL Managed Instance: tự động bảo trì, cho phép cấu hình hiều hơn so vơi SQL DB nhưng đòi hỏi nhiệm vụ administrative nhiều hơn
* Azure SQL VM: máy ảo có thiết lập SQL server, cho phép cấu hình toàn bộ.
* Azure SQL Edge: tối ưu hóa cho scenario IOT cần để làm việc với dữ liệu streaming

1. **Azure Database cho nhiều csdl quan hệ open-source:** MySQL, MariaDB, PostgreSQL
2. **Azure Cosmos DB:**

* Hệ thống Csdl global-scale non-relational (NoSQL) hỗ trợ nhiều APIs, cho phé lữu trữ và bảo trì dữ liệu dưới dạng JSON, key-value paires, column families, và graphs

1. **Azure Storage:** dịch vụ core cho phép lưu tữ dữ liệu trong:

* **Blob containers –** có thể mở rộng, lưu trữ với chí phí hiệu qua cho file nhị phân
* **File shares –** network file shares(vd: mạng trong công ty)
* **Tables –** lưu trữ key-value cho phép đọc ghi nhanh

1. **Azure Data Factory:**

* cho phép định nghĩa và lập lịch data pipeline để chuyển đổi và chuyển dạng dữ liệu.
* có thể tích hợp piplines với những dihcj vụ khác của Azure,
* cho phép nhập dữ liệu từ cloud, xử lý dữ liệu sử dụng điện toán dựa trên cloud
* được data engineer sử dụng để xây dụng giải pháp extract, transform, và load (ETL)

1. **Azure Synapse Analytics:** Giải pháp phân tích dữ liệu toàn diện và thống nhất, cung cấp một giao diện dịch vụ cho nhiều khả năng phân tích, bao gồm:

* Pipelines – cùng công nghệ với Azure Data Factory.
* SQL – csdl SQL có khả năng mở rộng cao, tối ưu hóa cho tải lượng dữ liệu trong warehouse
* Apache Spark – là một hệ thống xử lý dữ liệu phân tán mã nguồn mở, hỗ trợ nhiều dạng ngôn ngữ và APIs (Java, Scala, Python, SQL)
* Azure Synapse Data Explorer – giải pháp phân tích dữ liệu cho hiệu suất cao, được tối ưu hóa cho truy vấn thời gian thực của log, và sử dụng dữ liệu từ xa sử dụng Kusto Query Language (KQL)
* Data engineer sử dụng để tạo giải pháp phân tích dữ liệu thống nhất, kết hợp lấy dữ liệu pipelines, data warehouse, và data lake thông qua một dịch vụ duy nhất
* Data Analysis sử dụng SQL và Spark pools thông qua notebooks tương tác để khám phá và phân tích dữ liệu, và tận dụng sự tích hợp với các dịch vụ như Azure Machine Learning và Microsoft Power BI để tạo mô hình dữ liệu và xuất cái nhìn bên trong dữ liệu

1. **Azure Databricks**

* Phiên ản tích hợp Azure của Databricks platform,
* Kết hợp Apache Spark xử lý dư liệu với csdl SQL và tích hợp giao diện quản lý cho phép phân tích dữ liệu có khả năng mở rộng
* DE sử dụng Databricks và Spark skills để tạo kho dữ liệu phân tích lưu trữ trong Azure Databricks .
* DA sử dụng notebook hỗ trọ trong Azure Databricks đẻ truy vấn và hiển thị hóa dữ liệu một cách dễ dàng thông qua giao diện web.

1. **Azure HDInsight:** Azure HDInsight là một dịch vụ Azure cung cấp các cụm được lưu trữ trên Azure cho các công nghệ xử lý dữ liệu lớn nguồn mở phổ biến của Apache, bao gồm:

* Apache Spark – hệ thống xử lý dữ liệu phân tán, hỗ trọ nhiều ngôn ngữ và API,..
* Apache Hadoop – hệ thống phân tán sử dụng MapReduce để xử lý khối lượng lớn dữ liệu hiệu quả thông qua nhiều cụm nodes. Mapreduce có thể được viết bằng Java hoặc bởi giao diện như là Apache Hive – API dựa trên SQL chạy trên Hadoop
* Apache Hbase: một hệ thống mã nguồn mở cho lưu trữ NoSQL có khả năng mở rộng và truy vấn.
* Apache Kafka – tin nhắn broker cho xử lý dữ liệu dòng
* DE sử dụng Azure HDInsight để hỗ trợ phân tích dữ liệu lớn dựa trên nhiều công nghệ mã nguồn mở.

1. **Azure Stream Analytics:**

* Xử lý dòng thời gian thực, bắt những dòng dữ liệu từ input, và ứng dụng truy vấn để xuất và xử lý dữ liệu từ dòng dữ liệu input, và viết kết quả ra cho vieejv phân tích và xử lý sau này.
* DE sử dụng

1. **Azure Data Explorer:**

* Cung cấp truy vấn với hiệu suất cao cho log và dữ liệu và dữ liệu từ xa tương tự Azure Data Explorer
* DA sử dụng đẻ truy vấn và phân tích dữ liệu bao gồm một thuốc tính timestamp, như là thường được thấy trong file log và IoT dữ liệu từ xa.

1. **Microsoft Purview:**

* Giải pháp cho quản trị dữ liệu doanh nghiệp rộng
* Tạo map cho dữ liệu và track dữ liệu thông qua nhiều nguồn và hệ thống dữ liệu,
* Dữ liệu đáng tin cậy cho phân tích và báo cáo
* DA sử dụng để quản trjij dữ liệu thông qua doanh nghiệp và đảm bảo tính thống nhất dữ liệu được sử dụng trong phân tích

1. **Power BI:**

* Platform cho mô hình phần tích dữ liệu và báo cáo mà DA sử dụng để tạo và chia sẻ hiện thị hóa dữ liệu tương tác. Power BI report đựco tạo ra bằng cách sử dụng PowerBI Desktop.

1. **Explore relational data in Azure**
2. **Chuẩn hóa (normalization)**

* Giảm thiểu dữ liệu bị duplicate

1. Tách từng entity ra thành các bảng riêng
2. Tách thành những thuộc tính riêng lẻ (không gộp) thành từng cột
3. Primary key
4. Foreign key to link tables
5. **SQL**

* **Data Definition Language (DDL):** Create, Alter, Drop, Rename
* **Data Control Language (DCL):** Grant, deny, revoke
* **Data Manipulation Language (DML):** Select, Insert, Update, Delete

1. **View**

* Bảng ảo dựa trên kết quả của truy vấn Select
* Có thể truy vấn trong view, lọc dữ liệu tương tự như với table

1. **Store procedure**

* Định nghĩa SQL statements có thẻ chạy trên command.
* Đóng gói programmatic logic trong csdl cho những hành động mà ứng dụng cần để thujce thi khi làm việc với dữ liệu.
* Có thể sử dụng parameter trong store procedure

1. **Index (đã biết rồi)**
2. **Compare Azure SQL services**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **SQL Server Azure VMs** | **Azure SQL Managed Instance** | **Azure SQL Database** |
| **Type of cloud** | IaaS | PaaS | PaaS |
| **SQL compatibility (tương thích)** | Tương thích hoàn toàn với on-premises, dễ dàng lift and shift migrated | Tương thích gần 100%. Phần lớn on-premise database có thể chuyển lên chỉ cần thay đổi một số code | Hỗ trợ phần lớn khả năng của SQL Server. Một vài feature phụ thuộc vào ứng dụng on-premise có thể không available |
| **Kiến trúc** | SQL server instances được cài trong máy ảo.  Mỗi instance có thể hỗ trợ nhiều csdl | Each managed có thể hỗ tợ nhiều csdl.  Instance pools có thể được sử dụng để chia sẻ resources hiệu quả thông qua instance nhỏ hơn | Có thể cung cấp một csdl đơn như mong muốn, quản lý (logical) server, hoặc có thể sử dụng elastic pool (co giãn resource khi cần thiết -> tiết kiệm chi phí) để chia sẻ resources thông qua nhiều csdl và tận dụng khả năng mở rộng theo nhu cầu |
| **Available** | 99.99% | 99.99% | 99.995% |
| **Management** | Phải quản lý mọi khía cạnh của server, bao gồm: hệ điều hành, update SQL Server, cấu hình, backups, và bảo trì những tasks khác | Tự động cập nhật, backups, và recovery. Full control over security và resource allocation (cấp phát tài nguyên) cho csdl | Tự động cập nhật, backups, và recovery, auto-scale |
| **Usecase** | Sử dụng khi cần migrate hoặc mở rộng SQL Server on-premises và giữ toàn quyền quản lý | Sử dụng cho phần lớn cloud migration, cụ thể khi cần thay đổi một chút úng dụng đã tồn tại | Sử dụng để tạo ra một giải pháp cloud mới, hoặc migrate ứng dụng có ít instance-level dependencies |
| **Login** |  | With Azure Active Directory (AD)  Không cần cung cấp mỗi khi kết nối | Username and password  Cần cung cấp mỗi khi kết nối |
| **Others** | Nhanh chóng setup và chạy single SQL Server database | Tất cả các communication đều được mã hóa và signed using certificates | Support point-in-time restore cho phé khôi phục trạng thái csdl tại bất kỳ point nào trong quá khứ.  Được duplicate qua nhiều regions để cung cấp khả năng khôi phục  Advanced threat protection |

1. **Azure services for open-source databases**

* Có nhiều dịch vụ hỗ trợ cho những csdl quan hệ phổ biến, cho phép tổ chức chuyển dứng dụng từ on-premise lên Azure mà không cần nhiều sự thay đổi trong ứng dụng của họ
* Benefits of Azure Database for MySQL và MariaDB, pgSQL:

+ High availability features built-in.

+ Predictable performance.

+ Easy scaling that responds quickly to demand.

+ Secure data, both at rest and in motion.

+ Automatic backups and point-in-time restore for the last 35 days.

+ Enterprise-level security and compliance with legislation.

Uses pay-as-you-go

* Benefits of PGSQL:

+ highly available service

+ built-in failure detection and failover mechanisms

+ nếu quen với các tool sử dụng PG on-premises, continue to use this tool to connect to Azure Database for PostgreSQL

+ performing server backup and restore aren't available because server được quản lý và bảo trì bởi Microsoft

1. **Non-relational data**
2. **Azure blob storage**

* Cho phép lưu trữ khối lượng lớn các dữ liệu không có cấu trúc như là một ojects nhị phân lớn (or blobs) trên cloud
* Là cách thức hiệu quả để lưu trữ file dữ liệu trong format tối ưu cho lưu trữ dựa trên cloud và các ứng dụng có thể đọc và ghi bằng cách sử dụng Azure blob storage API dễ dàng.
* Trong Azre storage account, lưu trữ blobs trong *containers*. Container cung cấp cách thức hiệu quả cho việc nhóm các blobs liên quan lại với nhau. Có thể diều khiển ai có thể đọc và ghi blob trong container ở container level.
* Có thể tổ chức blobs dưới dạng có cấp bậc của các folders ảo, tương tự như file trong hệ thống file trên disk. Tuy nhiên, mặc định những folder này chỉ đươn giản là cách sử dụng ký tự / trong tên của bloc để tổ chức blobs. Folder hoàn toàn là ảo, và vì thế không thể thực hiện điều khiển truy cập hoặc thực hiện bulk.
* Có 3 loại blob khác nhau:

+ **Block blobs:**

* Block blob được xử lý như là một tập hợp các blocks.
* Mỗi block có thể thay đổi kich thước, lên đến 100MB.
* Mỗi block blob có thể chứa đến hơn 50000blocks, và có kích thước tối đa trên 4.7TB
* Phù hợp để lưu trữ những object riêng biệt, lớn, nhị phân thay đổi không thường xuyên

**+ Page blobs:**

* Được tổ chức với tập hợp những page có kích thước cố định 512-byte
* Được tối ưu hóa cho đọc ghi ngẫu nhiên. Có thể fetch và lưu trữ dữ liệu cho một trang khi cần
* Có thẻ lên đến 8TB dữ liệu
* Azure sử dụng page blobs để thực hiện disk ảo cho virtual machine

**+ Append blobs:**

* Một append blob là một block blob được tối ưu để hỗ trợ việc nối thêm.
* Chỉ có thể them vào cuối của một append blob, update và delete không được hỗ trợ.
* Mỗi block có thể thay đổi kích thước, lên đến 4 MB
* Kích thước tối đa của một append blob là 195GB
* 3 access tiers, giúp cân bằng độ trễ truy cập và chi phí lưu trữ:

+ hot tier: là mặc định, truy cập thường xuyên. Blob data được lưu trữ trên media có hiệu suất cao

+ cool tier: hiệu suất chậm hơn -> chi phí lưu trữ giảm hơn so với Hot tier. Sử dụng cho dữ liệu ít được truy cập. Thường cho việc mới tạo blob truy cập thường xuyên lúc đầu, nhưng sau đó giảm dần theo thời gian. Trong trường hợp này có thể tạo blob ở Hot, và sau đó migrate nó thành Cool. và bạn cũng có thể migrate từ cool sang hot.

+ Achive tier có chi phí lưu trữ thấp nhất, nhưng tang độ trễ. Thường cho những dữ liệu trong lịch sử không bị mất, nhưng yêu cầu truy xuất của những dữ liệu đó vô cùng thấp. lưu trữ hiệu quả trong trạng thái offline. Nếu như ở hot and cool độ trễ chỉ là vài mili giây, nhưng với Archive là vài tiếng. để truy xuất a blob trong Archive, bạn phải chuyển access tier to Hot hoặc Cool.

* Có thể tạo policies quản lý lifecycle cho blobs trong storage account. Lifecycle management policy có thể tự động chuyển blob từ Hot sang Cool và sau đó là Archive, khi thấy sử dụng ít thường xuyên. A lifecycle management policy có thể được sắp xếp để xóa những blob đã quá hạn.

1. **Azure DataLake Storage Gen 2**

* Gen 1: cho lưu trữ dữ liệu có cấp bậc cho data lake phân tích, được gọi là giải pháp big data analytical solutions làm việc với structured, semi-structured, unstructured dữ liệu được lưu trong file.
* Gen 2: mới hơn của service được tích hợp vào trong Azure Storage, cho phép tận dụng khả năng mở rộng của blob storage và quản lý chi phí của storage tiers
* Để tạo ra gen2 fils system, phải cho phép Hierarchical namespace của Azure Storage account:

+ có thể tạo ngay từ đầu

+ hoặc nâng cấp từ flat namespace nhưng khi đã nâng cấp thì không thể quay lại.

1. **Azure files**

* Cách thức để tạo cloud-based network shares, như bạn có thể tìm thấy tổ chức on-premises có thể tạo documents và những file khác available cho nhiều người.
* quản lý file shares trên Azure, tổ chức có thể giảm chi phí phần cứng, và chi phí duy trì, tận dụng high availability và khả năng mở rộng cloud storage cho files.
* Azure files có thể share lên tới 100 TB dữ liệu cho mỗi storage account
* Kích thước tối đa của một file dữ liệu lên đến 1TB
* Có thể hỗ trợ lên đến 2000 kết nối đồng thời trên mỗi file share
* Azure File Storage ofer 2 tiers:

+ standard tier: disk-base hardware trong datacenter

+ premium tier: solid-state disks -> greater throughput, nhưng được charge ở tỉ lệ cao hơn

* Hỗ trợ 2 phương thức network file sharing:

+ Server Message Block (SMB) thường được sử dụng qua nhiều hdh khác nhau (Windows, Linux, MacOS)

+ Network File System (NFS) được sử dụng bởi Linux và macOS, để tạo NFS, phải sử dụng premium tier storage account và tạo và cấu hình vnet thông qua đó, truycajap đến file được quản lý.

1. **Azure Tables**

* Là một giải pháp cho nosql, tận dụng bảng để lưu trữ key/value.
* Có thẻ được lưu trữ semi-structured
* Mỗi hàng phải có unique key (bao gồm khóa phân vùng và một khóa hàng)
* Khi điều chỉnh dữ liệu trong bảng, một cột timestamp sẽ lưu lại ngày giờ dữ liệu được thay đổi, ngoài ra những cột khác cũng có thể thay đổi.
* Không có khóa ngoại, quan hệ, stored procedures, views, hoặc những objects trong relational database.
* Dữ liệu trong bảng được chuẩn hóa xuống
* Để đảm bảo truy cập được nhanh hơn, chia bảng thành các phân vùng partition. Những hàng có cùng partition key sẽ được lưu trữ cùng nhau. Partioning không chỉ giúp tổ chức dữ liệu, mà nó còn cải thiện khả năng mở rộng và hiệu suất như sau:

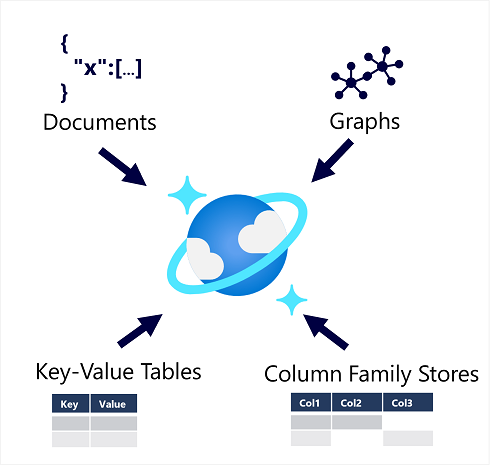
+ các phân mảnh là độc lập, có thể grow và shrink. Một bảng có thể có bao nhiêu phân mảnh cũng được.

+ Khi tìm kiếm dữ liệu, cần phải cung cấp partition key. Nó giúp giảm số lượng dữ liệu cần phải kiểm tra, và cải thiện hiệu suất bằng cách giảm IO

* Khóa trong Azure Table Storage bao gồm 2 thành phần: khóa partition cho biết ở partition nào trong bảng và khóa row cho biết row nào trong partition.
* Items được lưu trữ theo thứ tự khóa row.
* Cho phép ứng dụng truy xuất nhanh 1 hàng, hoặc một dãy các truy vấn các khối liên tiếp nahu trong partition

1. **Azure Cosmos DB**

* Hỗ trợ nhiều application programming interfaces (APIs) cho phép người lập trình có thể sử dụng ngữ nghĩa lập trình của nhiều loại lưu trữ dữ liệu phổ biến để làm việc với dữ liệu trong Cosmos DB database. Cấu trúc dữ liệu bên trong là trừu tượng, cho phép người lập tình sử dụng Cosmos DB để lưu trữ và truy vấn dữ liệu sử dụng APIs mà họ quen xài.



* Sử dụng chỉ mục và phân mảnh để hỗ trợ đọc ghi nhanh và có thể mở rộng cho khối lượng dữ liệu lớn.
* Có thể cho phép multi-region write, thêm Azure regions mà bạn chọn cho Cosmos DB account để có thể cung cấp toàn cầu cho người dùng có thể làm việc với dữ liệu tại bản sao địa phương của họ
* Khi nào sử dụng Cosmos DB:

+ Cosmos là một hệ thống quản lý csdl cso tính mở rộng cao. Tự động cấp phát không gian trong một container cho các partitions.

+ Mỗi partition có thể lên đến 10 GB kích thước.

+ chỉ mục được tạo và duy trì tự động.

+ không phải tốn chi phí administrative

+ cosmos được sử dụng bởi nhiều sản phẩm cho những ứng dụng quan trojgn trong việc mở rộng toàn cầu (Skype, Xbox, Microsoft 365, Azure, và nhiều thứ khác)

Cosmos phù hợp cho những scenario sau:

+ Iot và từ xa: những hệ thống này thường yêu cầu khối lượng lớn dữ liệu thường xuyên. Cosmos DB có thể chấp nhận và lưu trữ những thông tin này một cách nhanh chóng. Dữ liệu sau đó có thể được sử dụng cho những dịch vụ phân tích, ví dụ như Azure Machine Learning, Azure HDInsight, và PowerBI. Thêm vào đó có thể xử lý dữ liệu trong thời gian thực sử dụng Azure Functions được kích hoạt mỗi khi dữ liệu đến csdl.

+ Retail và marketing: Microsoft sử dụng Cosmos DB cho platform thương mại điện tử, nó chạy như là một phần của Windows Store và Xbox Live. Nó cũng được sử dụng trong ngành bán lẻ để lưu trữ catalog và cho tìm nguồn cung ứng sự kiện để có thể xử lý pipeline.

+ Gaming: csdl tier là một phần quan trọng trong các ứng dụng game. Csdl dành cho game gần phải nahnh và có thể xử lý các mức tăng đột biến về tỷ lệ yêu cầu trong thời gian ra mắt trò chơi mới và cập nhật tính năng

+ web and mobile applications: cosmos thường được sử dụng trong ứng dụng web và mobile, và cho những mô hình tương tác mạng xã hội, tích hợp dịch vụ bên thứ 3, và xây dựng lên trải nghiệm cá nhân phong phú.

1. **Azure Cosmos DB APIs**

* Là một csdl phân tán hoàn toàn được quản lý và không cần seerver cho ứng dụng ở bất kỳ size nào.
* Hỗ trợ cho cả relational và non-relational.
* Developer có thể xây dựng và chuyển giao các ứng dụng nhanh chóng sử dụng những csdl mã nguồn mở mà họ thích (PGSQL, MongoDB, và Apache Casandra).
* Khi cung cấp một Cosmos DB instance, bạn chọn csdl mà bạn muốn. Lựa chọn phụ thuộc vào:

+ kiểu dữ liệu được lưu trữ

+ những hỗ trọ cần thiết cho ứng dụng

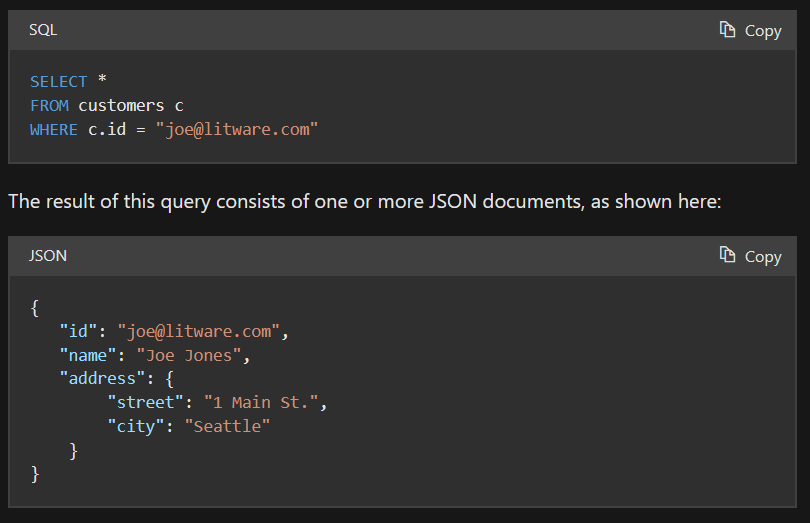
+ kỹ năng của các developer khi làm việc với dữ liệu được lưu trữ

* **Azure Cosmos Db for NoSQL:**

+ non-relational servicel làm việc với mô hình dữ liệu document

+ quản lý dữ liệu trong format JSON

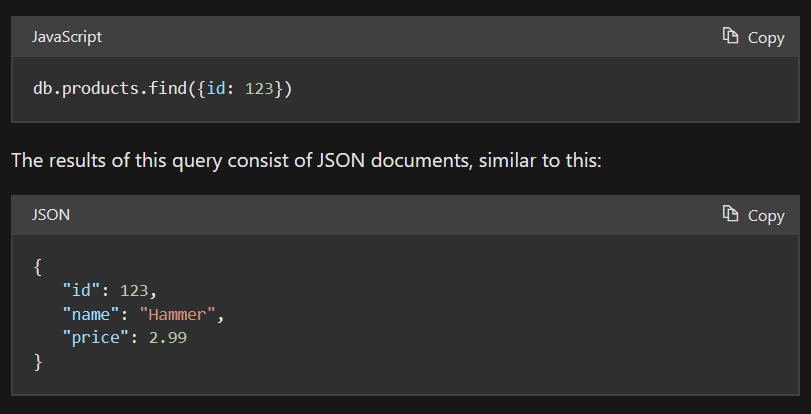
+ mặc dù là gải pháp cho NoSQL, nhưng sử dụng syntax SQL để làm việc với dữ liệu



* **Azure Cosmos DB for MongoDB:**

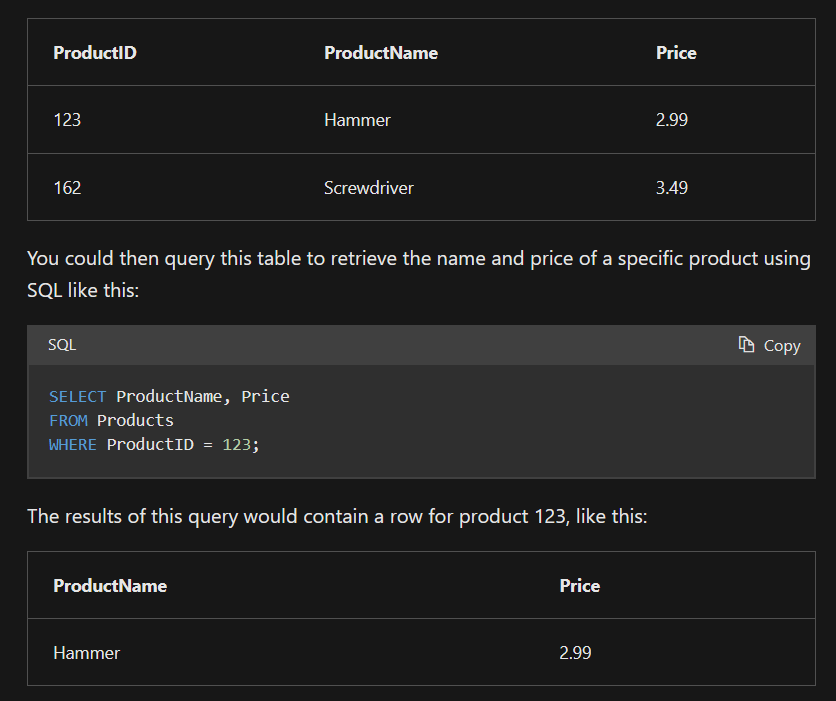
+ dữ liệu được lưu trữ trong Binary JSON (BSON) format. Azure cosmos DBfor MongoDb cho phé nhà phát triển sử dụng MongoDB client libraries và code để làm việc với dữ liệu trong Azure Cosmos DB.

+ MongoDB Query Language (MQL) sử dụng cú pháp gọn nhẹ, và hướng đối tượng trong đó nhà phát triển sử dụng objects để gọi methods



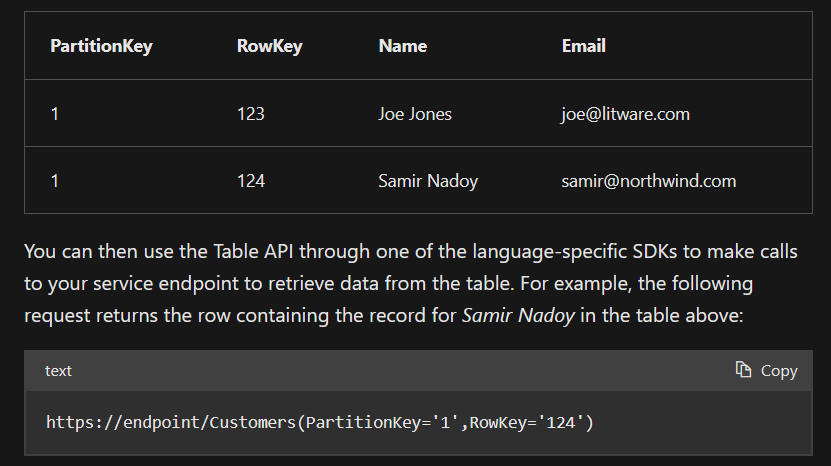
* **Azure Cosmos DB for PGSQL:**

+ PGSQL hệ cơ sở dữ liệu quan hệ phân tán toàn cầu, tự động phân mảnh dữ liệu để giúp xây dựng ứng dụng có độ mở rộng cao



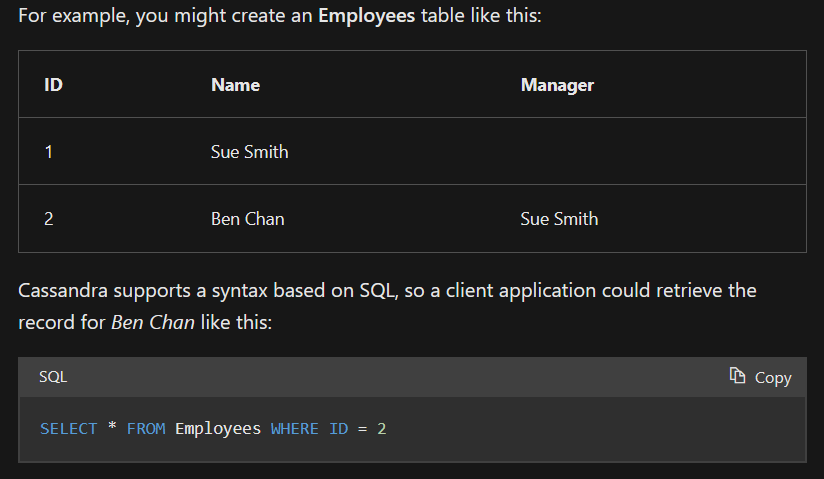
* **Cosmos DB for Table**

+ sử dụng đẻ làm việc với bảng key-value, tương tự như Azure Table Storage. Nó offers khả năng mở rộng và hiệu suất tốt hơn so với Azure Table Storage.



* **Azure Cosmos DB for Apache Cassandra**

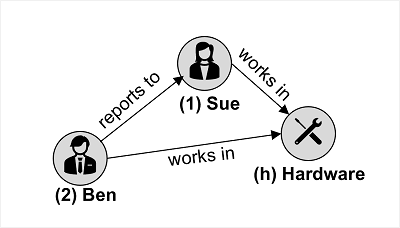
+ tương thích với Apache Cassandra, là một csdl mã nguồn mở sử dụng column-family storage structure. Column-familíes là các bảng, tương tự với csdl quan hệ, ngoại trừ việc nó không quan trojgn mỗi hàng phải có cùng số columns

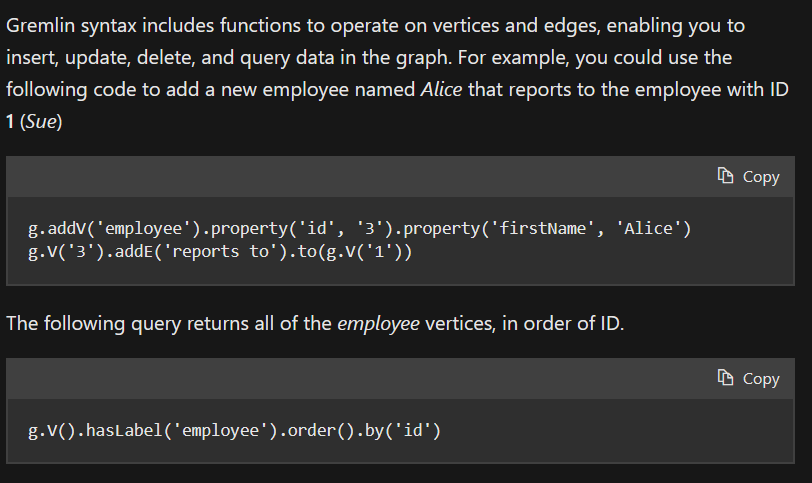


* **Azure Cosmos DB for Apache Gremlin:**

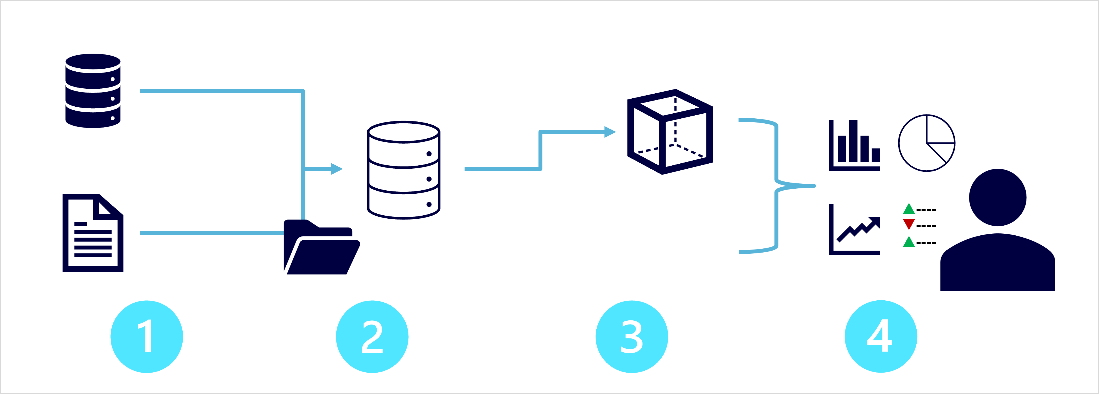
+ sử dựng với dữ liệu có structure đồ thị, trong đó, các thực thể được định nghĩa là các đỉnh hình thành nên các nodes.

+ mỗi nodes được kết nối bằng các cạnh mô tả mối quan hệ.

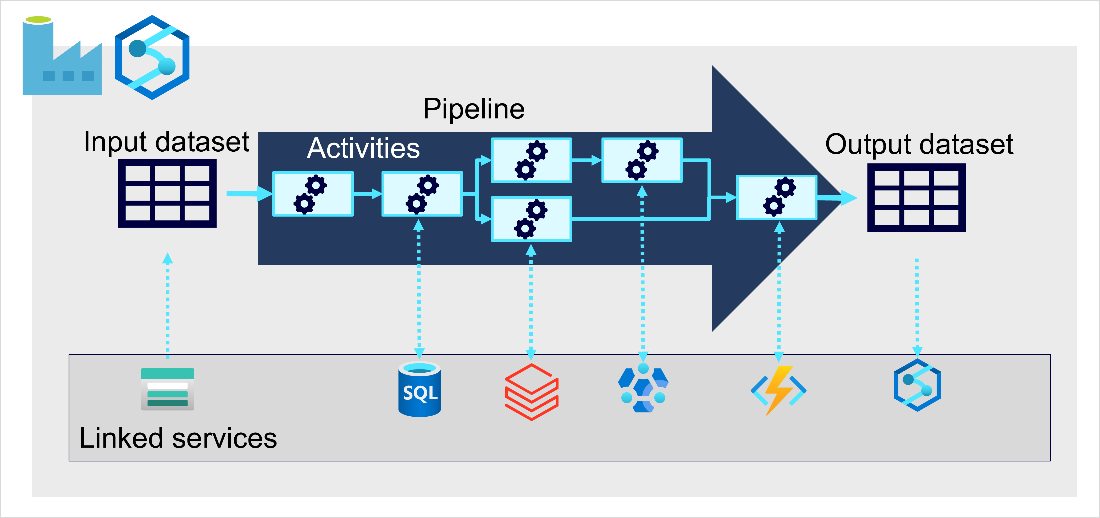




1. Data analytics
2. **Large-scale data warehousing**



1. Data ingestion and processing - dữ liệu từ một hay nhiều lưu trữ giao tác dữ liệu, files, streams thời gian thực, hoặc những nguồn khác được nạp vào data lake hoặc relational data warehouse. Có thể ELT hoặc ET. Mặc dù là cách nào thì kết quả cấu trúc dữ liệu được tối ưu hóa cho những truy vấn dùng để phân tích. Quá trình xử lý dữ liệu thường được hiện thực bởi hệ thống phân tán nó có thể xử lý khối lượng lớn dữ liệu một cách song song sử dụng multi-node clusster. Data ingestion gồm cả xử lý batch của dữ liệu tĩnh và real-time processing của streaming data.
2. Analytical data store - lưu trữ dữ liệu cho large scale analytics bao gồm dữ liệu quan hệ warehouse, file-system dựa trên data lakes, và kiến trúc hybid kết hợp giữa warehouse và data lake (data lakehouse / lake database)
3. Analytical data model – trong khi DA và DS có thể làm với dữ liệu trực tiếp từ Analytical data store, nó thường được tạo ra 1 hay nhiều mô hình dữu liệu mà nó tiinsh gọp trước dữ liệu để làm cho việc viết báo cáo, dashboard, trực quan hóa tương tác dễ dàng hơn. Mô hình đóng gói các mối quan hệ giữa các giá trị dữ liệu và dimensional entities để hỗ trợ drill-up/drill-down analysis
4. Data visualization – DA sử dụng dữ liệu từu analytical models, và trực tiếp từ analytical stores để tạo báo cáo, dashboards, và những hiển thị hóa khác. Thêm vào đó, người dùng trong tổ chức (ko chuyên về công nghệ) có thể tự thực hiện DA và reporting. Data visualization hiển trị trends, comparisons, và key performance indicators (KPIs) cho kinh doanh hoặc những tổ chức khác, và
5. **Data ingestion pipelines**



* Tạo và chạy piplines bằng cách sử dụng Azure Data Factory, hoặc Azure Synapse Analytics nếu muốn quản lý tất cả components của data warehouse solution trong một nơi thống nhất.
* Pipelinse bao gồm một hay nhiều hoạt động thực hiện trên data.
* Sử dụng linked services để nạp và xử lý dữ liệu -> sử dụng đúng công nghệ cho từng bước
* Cuối cùng có thể lưu trữ output dataset trong một linked service như Azure Sypnapse Analytics
* Pipeline có thể bao gồm những hành động được tích hợp sẵn, ko yêu cầu linked service.

1. **Analytical data stores**

* **Data warehouses:**

+ csdl quan hệ, dữ liệu lưu trữ trong schema tối ưu hóa cho DA hơn là transactional workloads.

+ dữ liệu từ transactional store được biến đổi thành schema, trong đó gia trị số học được lưu trong bảng, có thể liên quan đến 1 hay nhiều dimension table, nó biếu diễn các thực thể trong đó dữ liệu được tổng gộp.

+ Dimension table được gọi là star dimensions cso thể mở rộng thành snowflake

+ là một sự lựa chọn tốt cho transactionsal data được tổ chức thành lược đồ có cấu trúc của bảng, và sử dụng SQL để truy vấn.

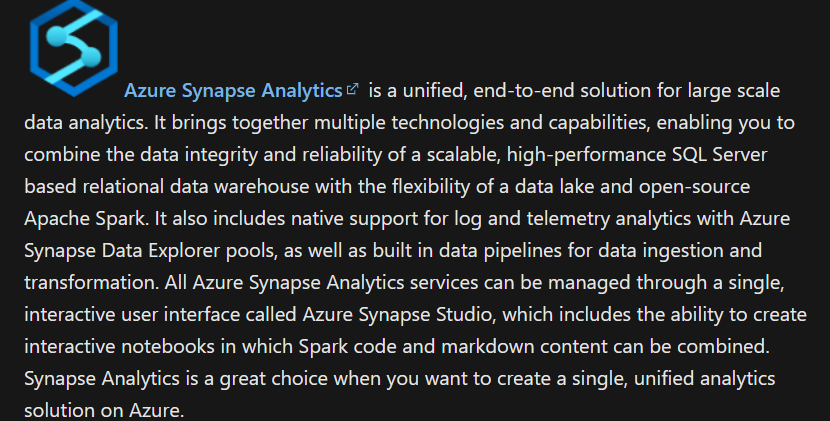
* **Data lakes:**

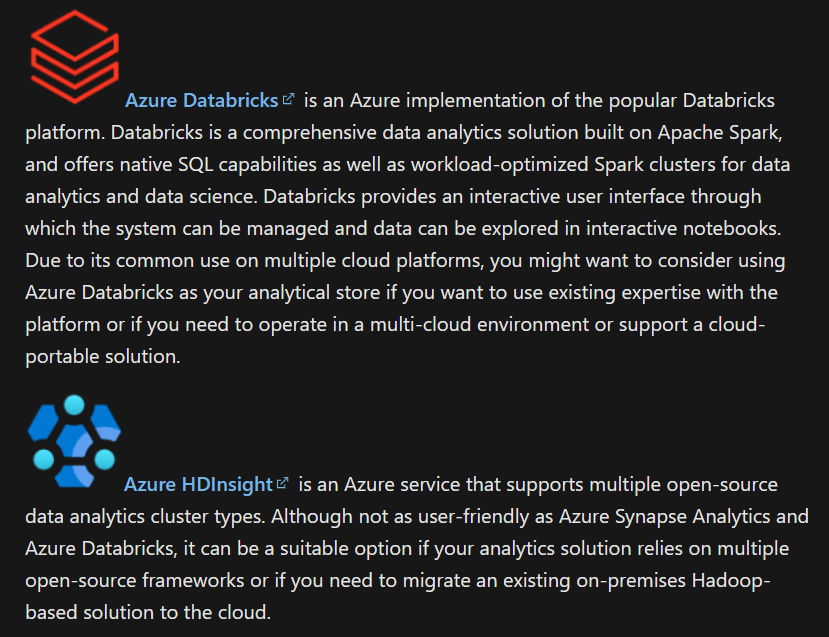
+ file store, thường là hệ thống file phân tán cho hiệu suất truy cạp dữ liệu cao.

+ hỗ trợ tố cho mix of structured, semi-structured, và thậm chí là unstructured

* **Hybrid approaches**

+ gồm những đặc điểm của datalakes và data warehouse





1. **Batch processing**

* Những phần tử dữ liệu đến được thu nhặt và lưu trữ, và toàn bộ nhóm được xử lý cùng nhau như là một bó. Có nhiều cách: xử lý dữ liệu dựa trên một khoảng thời gian đc lập lịch, hoặc có thể được kích hoạt khi có một khối lượng data đến, hoặc cũng có thể là kết quả của những sự kiện khác.
* Ví dụ: công ty thẻ ứng dụng xử lý billing. Khách hàng không nhận được bill mỗi khi sử dụng để mua mà chỉ nhận bill mỗi tháng (tổng mua sắm trong 1 tháng)
* **Advantage:**

+ khối lượng lớn dữ liệu được xử lý vào một khoảng thời gian thích hợp

Có thể lập lịch chạy khi máy tính hoặc hệ thống đang nhàn rỗi (giữa đêm, ngoài giờ cao điểm)

* **Disadvantage:**

+ thời gian trễ giữa lần nhập dữ liệu và lấy kết quả

+ tất cả dữ liệu dầu vào của batch phải được sẵn sàng trước khi batch được xử lý, nghĩa là dữ liệu phải được kiểm tra cẩn thận. Những vấn đề với dữ liệu, lỗi, chương trình ngưng thực thi có thể xảy tra trong suốt quá trình xử lý của batch. Dữ liệu đầu vào phải được kiểm tra cẩn thận trước khi có thể chạy lại. Chỉ cần một lỗi nhỏ trong dữ liệu có thể ngưng thực thi.

1. **Stream processing**

* Mỗi khi dữ liệu đến sẽ được xử lý.
* Data được xử lý độc lập trong thời gian thực hơn là xử lý thành bó trong 1 thời điểm.
* Có ích cho new, dữ liệu động được tạo ra liên tục
* Ví dụ:

+ theo dõi sự thay đổi của thị trường trong thời gian thực, tự động cân bằng danh mục đầu tư dựa trên giá cổ phần thay đổi.

+ online gamming

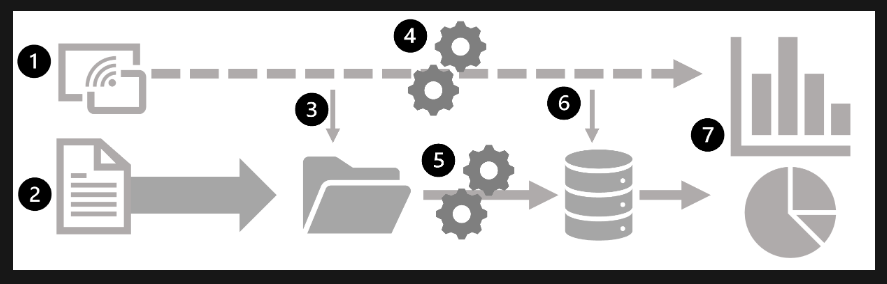
* lý tưởng cho những xử lý có liên quan đến thời gian. Ví dụ như: hệ thống giám sát tòa nhà cho việc khói thuốc và nhiệt độ để kích hoạt cảnh báo và mở cửa để cho cư dân trốn thoát ngay lập tức

1. **Different between batch and stream**

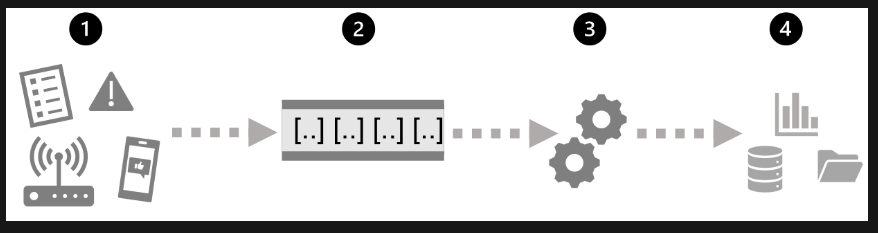
|  |  |  |
| --- | --- | --- |
|  | Batch | Stream |
| Data scope | Xử lý tất cả dữ liệu trong dataset | Chỉ truy cập đến những dữ liệu tức thì, hoặc trong một khoảng thời gian nhỏ (30s) |
| Data size | Hiệu quả cho datasets lớn | Thường cho từng record hoặc là những phần nhỏ batch bao gồm 1 vài records |
| Performance: latency là thời gian nhận dữ liệu và xử lý dữ liệu | Một vài giờ | Ngay lập tức, độ trễ thường vài giây đến vài mili giây |
| Analysis | Thực hiện những phân tích phức tạp | Những hàm, tính toán, tính gộp đơn giản |

1. **Combine batch and stream**

* Cho phép phân tích dữ liệu trong quá khứ và trong thời gian thực
* Sử dụng stream để bắt dữ liệu thời gian thực, xử lý nó bằng cách lọc hoặc tính gộp nó, và trình bày nó thông qua dashboard thời gian thực (vd: xem toàn bộ dữ liệu xe đi qua đường trong giờ hiện tại)
* Bảo quản kết quả được xử lý trong data store cho những phân tích lịch sử với xử lý batch (ví dụ phân tích kẹt xe trong năm vừa qua)



1. Dữ liệu từ nguồn streaming data được bắt trong thời gian thực
2. Dữ liệu từ những nguồn khác được nhập vào kho dữ liệu (thường là data lake) cho xử lý batch
3. Nếu không cần phân tích trong thời gian thực thì dữ liệu được bắt bởi stream được lưu vào trong data store cho những xử lý batch sau này
4. Khi cần phân tích thời gian thực, công nghệ bộ xử lý stream được sử dụng để chuẩn bị cho phân tích streaming data hoặc để hiển thị, thường bằng cách filter hoặc tính gộp
5. Những dữ liệu không streaming đinh kỳ được xử lý batch để chuẩn bị cho phân tích, và kết quả được duy trì trong kho lưu trữ dữ liệu phân tích (thường là data warehouse) cho phân tích lịch sử.
6. Kết quả của xử lý stream có thể được duy trì trong kho lưu trữ dữ liệu phân tích để hỗ trợ phân tích lịch sử
7. Các tools phân tích và hiển thị được sử dụng để trình bày và khám phá dữ liệu thời gian thực cũng như trong lịch sử
8. **Stream processing architecture**

****

1. Sensor, social media message, log file entry being written
2. Dữ liệu sinh ra được bắt bởi streaming source để xử lý. Có thể là folder trong kho lưu trữ dữ liệu cloud hoặc là bảng trong csdl.

Source có thể là hàng đợi -> đóng gói luận lý để bảo đảm dữ liệu được xử lý theo thứ tự và mỗi sự kiện chỉ được xử lý một lần.

1. Sự kiện dữ liệu được xử lý, thường là những câu truy vấn vĩnh viễn (vd: tính toán số lượng của cảm biến khí thải mỗi phút)
2. Kết quả của xử lý stream được viết thành output (or sink), nó có thể là file, bảng csdl, dashboard thời gian thực, hoặc hàng đợi khác cho những xử lý truy vấn xuôi dòng tiếp theo.

* **Real-time analytics in Azure**

+ **Azure Stream Analytics**: PaaS solution để bạn có thể sử dụng để định nghĩa *Streaming jobs* nó lấy dữ liệu từ streaming source, áp dụng những câu truy vấn vĩnh viễn, và viết kết quả vào output.

+ **Spark Structured Streaming**: thư viện mã nguồn mở cho phép phát triển những giải pháp streaming phức tạp trên Apache Sparck dựa trên dịch vụ, bao gồm Azue Synapse Analytics, Azure Databricks và Azure HDInsight.

+ **Azure Data Explorer**: là csdl hiệu suất cao và dịch vụ phân tích tối ưu cho việc ingesting và querying batch hoặc streaming dữ liệu với thành phần time-series, và có thể sử dụng như là dịch vụ Azure độc lập hoặc như là Azure Synapse Data Explorer

* **Source for stream processing**

+ Azure Event Hubs: dịch vụ nhập dữ liệu, có thể sử dụng để quản lý hàng đợi của dữ liệu sự kiện, đảm bỏa mỗi sự kiện được xử lý theo đúng thứ tự và xử lý 1 lần duy nhất

+ Azure IoT Hub: dịch vụ nhập dữ liệu, tương tự với Azure Event Hubs, nhưng được nối ưu hóa để quản lý dữ liệu sự kiện từ thiết bị Iot.

+ Azure Data Lake store Gen 2: dịch vụ kho lưu trữ có khả năng mở rộng cao, thường được sử dụng trong xử lý batch, nhưng cũng có thể dùng như nguồn của dữ liệu streaming.

+ Apache Kafka: giải pháp nhập dữ liệu mã nguồn mở, được sử dụng với Apache Spark. Có thể sử dụng Azure HDInsight để tạo cụm Kafka.

* **Sink for stream processing**

+ Azure Event Hubs: sử dụng để sắp xếp hàng đợi dữ liệu đã được xử lý cho những xử lý downstream sau này.

+ Azure data Lake Store Gen 2 hoặc Azure blob storage: được sử dụng để duy trì kết quả được xử lý như là 1 file.

+ Azure SQL database hoặc Azure Synapse Analytics hoặc Azure Databricks: được sử dụng để duy trì kết quả đã được xử lý trong bảng csdl cho truy vấn và phân tích

+ Microsoft Power BI: tạo hiển thị dữ liệu thời gian thực trong báo cáo và trong dashboard.

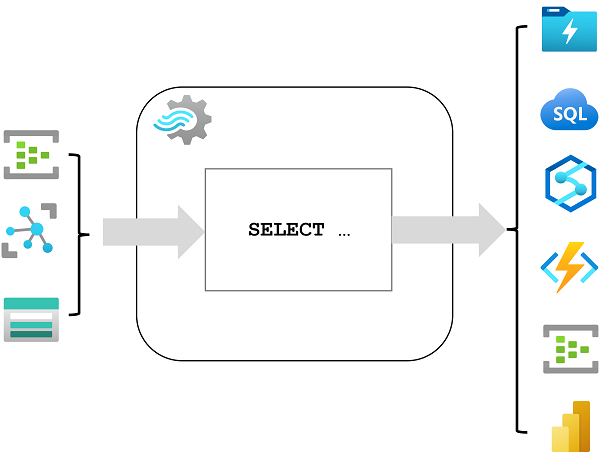
1. **Azure Stream Analytics**

* Dịch vụ cho xử lý những sự kiện phức tạp và phân tích dữ liệu stream. Được sử dụng để:

+ nhập dữ liệu từ input, như là Azure event hub, Azure IoR Hub, Azure storage blob container.

+ xử lý dữ liệu bằng cách sử dụng query để select, project, và tính gọp giá trị dữ liệu.

+ viết kết qua ra output, như là Azure DataLake Gen 2, Azure SQL Database, Azure Synapse Analytics, Azure Functions, Azure event hub, Microsoft Power BI, hoặc những cái khác.



* Một khi bắt đầu, stream analytics sẽ chạy vĩnh viễn, xử lý dữ liệu mới khi nó đén tỏng input và lưu trữ kết quả trong output.
* Là một lựa chọn công nghệ tốt khi cần bắt dữ liệu liên tục từ streaming source, filter hoặc aggregate it, và gửi kết quả đến data store hoặc downstrend xử lý cho phân tích và báo cáo.

1. **Apache spark on Microsoft Azure**

* Một processing framework phân tán cho phân tích dữ liệu mở rộng lớn, có thể sư dụng Spark trong những dịch vụ sau:

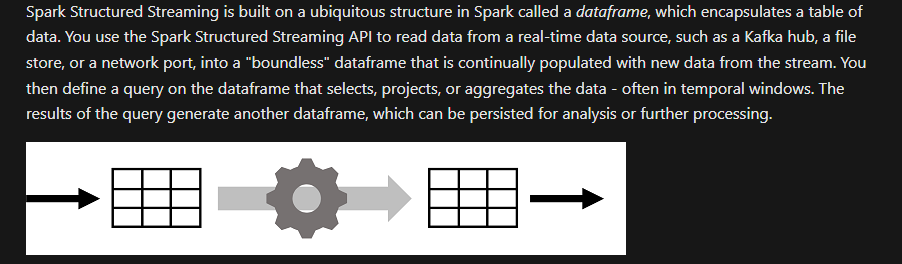
+ Azure Synapse Analytics

+ Azure Databricks

+ Azure DHInsight

* Được sử dụng để chạy code (python, scala, java) đòng thời qua nhiều cụm nodes, cho phép xử lý khối lượng lớn dữ liệu hiệu quả. Có thể sử dụng cho cả batch và stream
* **Spark Structured Streaming:**

+ sử dụng thu viện spark Structured streaming, cng cấp giao diện API đẻ nhập dữ liệu, xử lý và xuất ra kết quả từ một stream data



+ lựa chọn tố cho phân tích thời gian thực khi bạn cần kết hợp streaming data vào trong Sparrk dựa trên data lake hoặc kho lưu trữ dữ liệu phân tích.

* **Delta Lake:**

+ storage layer mã nguồn mở thêm hỗ trợ cho tính nhất quán của giao tác, thực thi lược đồ, và những đặc điểm thông dụng khác của data warehouse. Nó cũng thống nhất lưu trữ cho streaming và batch, và có thể sử dụng trong spark đẻ định nghĩa relational tables cho cả batch và stream.

+ khi sử dụng cho stream processing, Delta Lake table có thể được sử dụng như là streaming source cho truy vấn dữ liệu thời gian thực, hoặc như là sink nơi mà stream of data được viết.

+ thời gian chạy Spark trong Azure Synapse Analytics và Azure Databricks bao gồm hỗ trợ cho Delta Lake

+ kết hợp với Spark Structured Streaming là một lựa chọn tốt khi cần trừu tượng batch và stream processed data trong dtaa lake đăng sau lược đồ quan hệ cho SQL-based querying và phân tích.

1. **PowerBI tools và workflow**

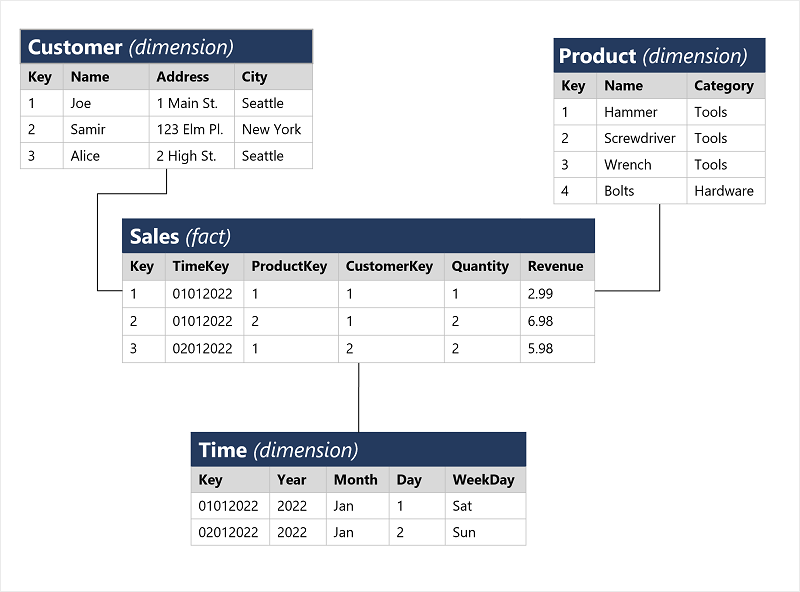
**A picture containing text, screenshot, computer icon, software

Description automatically generated**

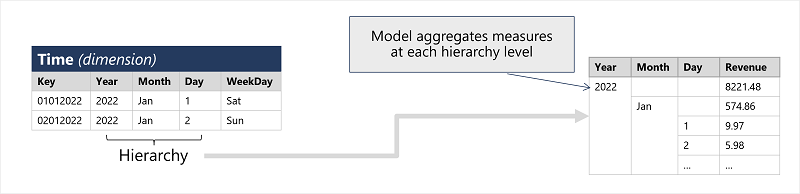
* Nhập dữ liệu từ nhiều nguồn dữ liệu, kết hợp và tổ chứ dữ liệu từ những nguồn đó thành các mô hình dữ liệu phân tích, và tạo báo cáo chứa những hiển thị có thể tương tác được với dữ liệu
* Sau khi tạo mô hình dữ liệu và báo cáo, publish lên PowerBI service - dịch vụ cloud, trong đó báo cáo có thể published và tương tác với những khách hàng business.
* Có thể sử dụng những mô hình đơn giản và điều chỉnh báo cáo trực tiếp trên dihcj vụ sử dụng web browser, nhưng chức năng này có một vài giới hạn so với Power BI desktop
* Sử dụng dịch vụ để định thời gian làm mới nguồn dữ liệu mà báo cáo sử dụng, và để chia sẻ báo cáo với những người dùng khác.

1. **Concept of data modeling**

* Table and schema:



* Attribute hierarchies



* Analytical modeling in Power BI:

+ định nghĩa mô hình phân tích từ các bảng dữ liệu, có thể nhập vào từ 1 hoặc nhiều nguồn.

+ sử dụng giao diện mô hình hóa dữ liệu để định nghĩa mô hình phân tích bằng cách tạo ra mối quan hệ giữa fact và dimension tables, xác định hierarchies, xác định kiểu dữ liệu và hiển thị formats cho các trường trong bảng, và quản lý những thuộc tính khác của dữ liệu của bạn giúp cho định nghĩa mô hình phân tích giàu hơn để phân tích

1. **Considerations for data visualization:**

* Có nhiều cách hiển thị dữ liệu:

+ bảng/text

+ bar, column charts

+ Line charts

+ Pie charts

+ Scatter plots

+ Maps

+ interactive report in Power BI