**DATA WAREHOUSE**

**1. Relational Data Warehouse (DW)**

**\*Data Warehouse là gì?**

* Là hệ thống lưu trữ và phân tích dữ liệu, được xây dựng để hỗ trợ việc ra quyết định trong tổ chức.
* Relational DW: sử dụng Cơ sở dữ liệu quan hệ (RDBMS) để lưu trữ dữ liệu.

**\*Đặc điểm của Relational DW:**

* **Subject-oriented (Hướng chủ đề)**
* Dữ liệu được tổ chức xung quanh **chủ đề chính** như: khách hàng, sản phẩm, doanh thu…
* Không tập trung vào quá trình xử lý giao dịch, mà là để phân tích tổng hợp theo chủ đề.
* **Integrated (Tích hợp)**
* Dữ liệu đến từ nhiều nguồn khác nhau (**database, file, API,...**) được chuẩn hóa và tích hợp về **định dạng thống nhất**.
* Ví dụ: dữ liệu ngày tháng được đưa về cùng một định dạng YYYY-MM-DD.
* **Time-variant (Chuỗi thời gian)**
* Dữ liệu có **dấu thời gian** đi kèm, lưu lại lịch sử thay đổi qua các mốc thời gian khác nhau.
* Giúp phân tích xu hướng, ví dụ: doanh số theo tháng, năm...
* **Non-volatile (Không biến đổi)**
* Sau khi dữ liệu đã nạp vào kho thì **không sửa, không xóa**, chỉ có thêm mới.
* Đảm bảo tính nhất quán và phục vụ truy vết lịch sử.
* **Summarized (Tóm tắt)**
* Dữ liệu có thể được **tổng hợp lại** để tăng hiệu suất truy vấn, ví dụ:
  + Tổng doanh thu theo quý.
  + Số đơn hàng mỗi tháng.
* Giúp trả lời nhanh các câu hỏi dạng BI.
* **Not normalized (Không được chuẩn hóa)**
* Khác với OLTP, DW thường **phi chuẩn hóa** để giảm số lượng bảng, tăng tốc truy vấn.
* Ví dụ: bảng Fact chứa thông tin đã gộp nhiều chiều, không tách nhỏ ra nhiều bảng con.
* **Metadata (Siêu dữ liệu)**
* Là dữ liệu mô tả về dữ liệu khác: định nghĩa bảng, kiểu dữ liệu, mô tả các cột,...
* Giúp hiểu rõ cấu trúc DW, dùng trong quản trị và phân tích.
* **Web based, relational/multi-dimensional**
* DW có thể sử dụng nền tảng Web để truy cập.
* Dữ liệu được lưu trong mô hình **quan hệ** hoặc **đa chiều** (multidimensional – ví dụ OLAP cubes).
* **Client/server, real-time/right-time/active**
* DW hiện đại hỗ trợ các mô hình:
  + **Client/server**: phân tán giữa máy chủ và máy trạm.
  + **Real-time**: cập nhật dữ liệu liên tục.
  + **Right-time**: cập nhật đúng lúc cần.
  + **Active**: tương tác trực tiếp với các hệ thống phân tích/kết quả.

**\*So sánh với OLTP (Online Transaction Processing):**

| Đặc điểm | OLTP | Data Warehouse |
| --- | --- | --- |
| Mục đích | Xử lý giao dịch | Phân tích dữ liệu |
| Thiết kế | Tối ưu hóa ghi/đọc nhanh | Tối ưu hóa truy vấn phức tạp |
| Dữ liệu | Hiện tại, chi tiết | Lịch sử, tổng hợp |
| Cấu trúc | Chuẩn hóa (Normalization) | Phi chuẩn hóa (Denormalization) |

**2. The Top-Down Approach (Phương pháp từ trên xuống)**

**\* Định nghĩa:**

Top-down approach là phương pháp xây dựng Data Warehouse bắt đầu từ một kho dữ liệu tổng thể (enterprise-wide) rồi sau đó mới tạo ra các data mart (kho dữ liệu con) cho các bộ phận như kế toán, marketing, bán hàng...

\***Chi tiết các bước**

**(1) Understand corporate strategy**

**Hiểu chiến lược doanh nghiệp**

* Xác định chiến lược, mục tiêu, giả thuyết.
* Trả lời câu hỏi: *Doanh nghiệp muốn biết điều gì từ dữ liệu?*
* Ví dụ: “Làm sao để tăng doanh thu quý sau?” hay “Những sản phẩm nào đang giảm sức mua?”

**(2) Define the business requirements**

**Xác định yêu cầu nghiệp vụ**

* Hiểu rõ các **mục tiêu, chỉ số hiệu suất (KPIs)**, thông tin mà các phòng ban cần.
* Phân tích nhu cầu của từng nhóm người dùng (marketing, bán hàng, tài chính,...)
* Cũng có thể xem như là phân tích yêu cầu báo cáo.

**(3) Design the data warehouse architecture**

**Thiết kế kiến trúc của DW**

* Dựa vào yêu cầu ở bước 2, vẽ ra **kiến trúc cấp cao** cho kho dữ liệu.
* Bao gồm:
  + Cấu trúc dữ liệu (schema)
  + Mô hình dữ liệu (data models)
  + Quy trình tích hợp dữ liệu

**(4) Develop the data model**

**Xây dựng mô hình dữ liệu chi tiết**

* Thiết kế **các thực thể (entities)** và **mối quan hệ (relationships)**.
* Xác định mức độ chi tiết (granularity): ví dụ lưu dữ liệu theo ngày, tuần hay tháng?
* Giai đoạn này cần rất sát với thực tế nghiệp vụ.

**(5) Build the appropriate database, schemas, tables, and fields**

**Tạo CSDL, lược đồ, bảng và trường dữ liệu**

* Bắt đầu xây dựng **cấu trúc vật lý** cho kho dữ liệu:
  + DBMS (ví dụ: Oracle, SQL Server,...)
  + Schema (dạng star, snowflake)
  + Bảng Fact và Dimension
* Đây là phần “viết dữ liệu lần đầu” (schema-on-write).

**(6) Data extraction, transformation, and loading (ETL)**

**ETL – Trích xuất, chuyển đổi và nạp dữ liệu**

* Dùng các công cụ ETL để:
  + **Extract**: trích xuất dữ liệu từ nguồn (CSDL vận hành, file Excel,...)
  + **Transform**: chuẩn hoá, làm sạch, đổi format.
  + **Load**: nạp vào kho dữ liệu.

**3. Benefits and Drawback**

**\* Lợi ích khi sử dụng Relational Data Warehouse (RDW)**

**Hiệu suất & tối ưu hệ thống**

* **Giảm áp lực lên hệ thống sản xuất**  
  (DW là môi trường riêng, không ảnh hưởng hệ thống vận hành OLTP).
* **Tối ưu cho truy cập đọc**  
  Truy vấn nhanh hơn, thích hợp với các câu hỏi phân tích lớn.

**Tích hợp & phân tích dữ liệu**

* **Tích hợp nhiều nguồn dữ liệu khác nhau**  
  (kế toán, bán hàng, kho, marketing, v.v.)
* **Chạy các báo cáo lịch sử chính xác**  
  Nhờ dữ liệu có yếu tố thời gian (time-variant).
* **Cấu trúc dữ liệu tốt hơn**:  
  Dễ dàng đổi tên bảng, làm rõ mối quan hệ.

**Quản lý & bảo mật dữ liệu**

* **Bảo vệ khi hệ thống ứng dụng nâng cấp**  
  (Dữ liệu DW không bị ảnh hưởng nếu app thay đổi).
* **Giảm lo ngại về bảo mật**  
  DW có thể quản lý quyền truy cập, dữ liệu nhạy cảm tốt hơn.

**Khả năng lưu trữ & chất lượng dữ liệu**

* **Lưu trữ dữ liệu lịch sử**
* **Quản lý dữ liệu master tốt hơn (MDM)**
* **Cải thiện chất lượng dữ liệu**  
  Nhờ phát hiện lỗi hoặc lỗ hổng khi so sánh giữa các hệ thống nguồn.

**Tiện lợi cho người dùng**

* **Không cần IT tạo báo cáo liên tục**  
  Người dùng tự thao tác được với dashboard/báo cáo.

**Hạn chế khi dùng Relational DW**

**Chi phí và độ phức tạp**

* **Phức tạp (Complexity)**: thiết kế, xây dựng, bảo trì đều đòi hỏi chuyên môn cao.
* **Chi phí cao (High costs)**: phần cứng, phần mềm, đội ngũ nhân sự.

**Vấn đề hiệu năng và độ trễ**

* **Biến đổi dữ liệu tốn thời gian (ETL)**
* **Độ trễ dữ liệu (Data latency)**: không phù hợp với hệ thống cần dữ liệu thời gian thực.
* **Khung bảo trì hệ thống (Maintenance window)**

**Tính linh hoạt hạn chế & bảo mật**

* **Tính linh hoạt thấp hơn** so với các mô hình mới (như Data Lake, Data Mesh).
* **Lo ngại về quyền riêng tư và bảo mật** nếu thiết kế kém.

**4. Nạp dữ liệu vào DW**

Đây là quá trình thu thập, xử lý, chuyển đổi và tải dữ liệu từ các hệ thống nguồn (như cơ sở dữ liệu giao dịch, CRM, ERP, Excel, v.v.) vào kho dữ liệu (DW).

**Các bước chính trong quá trình này:**

1. Extract (Trích xuất)

* Lấy dữ liệu từ các nguồn khác nhau: hệ thống giao dịch, cơ sở dữ liệu phân tán, file Excel,...
* Có thể trích xuất toàn bộ hoặc từng phần (incremental).

2. Transform (Biến đổi)

* Là bước xử lý và làm sạch dữ liệu:
  + Chuẩn hóa định dạng (ví dụ: ngày/tháng/năm)
  + Xử lý giá trị thiếu, lỗi logic
  + Tính toán lại các giá trị
  + Tích hợp các nguồn dữ liệu lại với nhau
* Đảm bảo dữ liệu đúng – sạch – nhất quán trước khi nạp vào DW.

3. Load (Tải vào)

* Dữ liệu đã xử lý sẽ được nạp vào kho dữ liệu.
* Có 2 kiểu:
  + Full load: nạp lại toàn bộ dữ liệu
  + Incremental load: chỉ nạp dữ liệu mới hoặc thay đổi (tối ưu hơn)

🔧 Công cụ hỗ trợ quá trình này gọi là ETL tools, ví dụ:

* Informatica
* Talend
* Microsoft SSIS
* Apache Nifi
* Airbyte, Fivetran (cho Cloud DW)

**5. Những component khác của DW**

1. **ODS – Operational Data Store (Kho dữ liệu vận hành tạm thời)**

**Đặc điểm:**

* **Chứa dữ liệu gần với thời gian thực** (near real-time).
* Là nơi **lưu trữ tạm thời** dữ liệu giao dịch mới phát sinh.
* Dữ liệu trong ODS **chưa được tổng hợp hay chuẩn hóa hoàn toàn**, chủ yếu dùng cho báo cáo nhanh/ngắn hạn.

**🔧 Dùng khi nào?**

* Khi cần **phân tích nhanh các thay đổi mới nhất** trong hệ thống (ví dụ: số đơn hàng hôm nay, số khách đăng ký mới).
* Trước khi chuyển dữ liệu vào kho dữ liệu chính (EDW), thường sẽ đi qua ODS.

1. **Data Mart (Chợ dữ liệu)**

**Đặc điểm:**

* Là **một phần nhỏ** của Data Warehouse, dùng riêng cho **một bộ phận cụ thể** như Marketing, Sales, Nhân sự,...
* Dữ liệu trong Data Mart được **lọc, rút gọn và tổ chức phù hợp với nhu cầu phân tích** của từng nhóm người dùng.

**Ví dụ:**

* **Sales Data Mart**: chỉ chứa dữ liệu về doanh số, khách hàng, khu vực bán hàng.
* **HR Data Mart**: chứa thông tin nhân viên, lương, ngày công,...

1. **EDW – Enterprise Data Warehouse (Kho dữ liệu doanh nghiệp)**

**Đặc điểm:**

* Là **kho dữ liệu trung tâm** – nơi chứa dữ liệu đã được trích xuất, chuyển đổi, làm sạch và tích hợp từ nhiều nguồn.
* Dữ liệu trong EDW là **dữ liệu tĩnh, ổn định, có tổ chức rõ ràng**.
* Phục vụ cho các nhu cầu phân tích tổng thể, báo cáo chiến lược dài hạn.

**Vai trò:**

* Là **trái tim** của toàn bộ hệ thống DW.
* Các **ODS và Data Marts thường lấy dữ liệu từ đây** (sau khi xử lý ETL xong).

1. **Metadata (Siêu dữ liệu)**

**Đặc điểm:**

* Metadata là **“dữ liệu về dữ liệu”** – giúp bạn hiểu được:
  + Dữ liệu này đến từ đâu?
  + Nó có ý nghĩa gì?
  + Nó được xử lý như thế nào?

**Có 2 loại metadata chính:**

* **Technical Metadata**: thông tin kỹ thuật – như tên bảng, kiểu dữ liệu, mối quan hệ giữa các bảng.
* **Business Metadata**: định nghĩa nghiệp vụ – như "doanh thu ròng là gì", "một khách hàng tiềm năng được tính ra sao",...

**Tác dụng:**

* **Hỗ trợ việc quản lý dữ liệu, truy vết, kiểm tra chất lượng**.
* Là “bản đồ” để nhà phân tích/truy vấn không bị lạc giữa biển dữ liệu.

**6. DW Framework/Architectures**

**Kiến trúc phổ biến của Data Warehouse gồm:**

* **Single-tier architecture**: ít dùng vì hiệu suất thấp, dữ liệu tập trung một nơi.
* **Two-tier architecture**: tách riêng tầng lưu trữ dữ liệu và tầng trình bày (presentation).
* **Three-tier architecture** (chuẩn nhất):
  1. **Tầng nguồn dữ liệu**: các hệ thống OLTP (Operational).
  2. **Tầng Data Warehouse**: lưu trữ và xử lý dữ liệu (ETL).
  3. **Tầng truy cập dữ liệu**: công cụ phân tích, truy vấn, dashboard.

**7. Data Warehouses Development**

**🔧 Các bước phát triển DW gồm:**

1. **Phân tích yêu cầu nghiệp vụ**.
2. **Thiết kế kiến trúc hệ thống** (chọn Top-down hay Bottom-up).
3. **Xác định nguồn dữ liệu**.
4. **Phát triển ETL pipeline**.
5. **Thiết kế mô hình dữ liệu** (schema).
6. **Kiểm thử, triển khai và vận hành**.

**8. Dimensional Modeling in DW**

**Là kỹ thuật mô hình hóa dữ liệu phổ biến cho DW:**

* Gồm **Fact table (bảng sự kiện)** và **Dimension tables (bảng chiều)**.
* Các mô hình chính:
  + **Star Schema**: bảng fact ở giữa, liên kết trực tiếp với các bảng dimension.
  + **Snowflake Schema**: bảng dimension được chuẩn hóa thêm 1-2 cấp.

**9. Feeding Data into the Warehouse**

**Các phương pháp đưa dữ liệu vào DW:**

\* ETL (Extract – Transform – Load):

* Extract: Trích xuất dữ liệu từ hệ thống nguồn.
* Transform: Làm sạch, chuẩn hóa, chuyển đổi dữ liệu.
* Load: Đưa dữ liệu đã xử lý vào DW.

Khi nào dùng ETL?

* Khi nguồn và DW khác công nghệ (ví dụ từ file Excel/CSV → Oracle).
* Khi cần xử lý dữ liệu phức tạp bên ngoài DW.
* Khi dữ liệu vừa và nhỏ, xử lý ngoại vi là phù hợp.

Công cụ:  
Informatica, Talend, DataStage, Pentaho, AWS Glue...

\* ELT (Extract – Load – Transform):

* Extract: Trích xuất dữ liệu.
* Load: Đưa dữ liệu thô vào DW trước.
* Transform: Xử lý bên trong DW bằng SQL hoặc công cụ native.

Khi nào dùng ELT?

* Khi nguồn và DW tương thích (cùng RDBMS).
* Dữ liệu lớn cần xử lý nhanh.
* DW có sức mạnh xử lý cao (cloud DW như Snowflake, BigQuery).

Cách lựa chọn giữa ETL và ELT phụ thuộc vào:

1. Độ phức tạp của các bước transform.
2. Tốc độ yêu cầu truy cập dữ liệu phân tích.
3. Công nghệ nguồn và đích có tương thích nhau không.

**10. Analysis of Data in DW**

**Sau khi dữ liệu vào DW, bạn có thể:**

* **Tạo báo cáo (report)**.
* **Xây dựng dashboard** (Power BI, Tableau...).
* **Khai phá dữ liệu (data mining)**.
* **Dự báo, mô hình hóa (predictive analytics)**.
* Hỗ trợ **BI (Business Intelligence)** và ra quyết định.

**11. The Death of the Relational Data Warehouse**

**Vì sao DW truyền thống đang dần lỗi thời:**

* **Big Data** và dữ liệu phi cấu trúc (unstructured) ngày càng nhiều.
* **Cloud-based DW** (ví dụ: Snowflake, BigQuery, Redshift) linh hoạt hơn.
* Nhu cầu **real-time analytics** khiến ETL truyền thống không đủ nhanh.
* **Data Lake + Lakehouse** trở thành giải pháp thay thế DW.

**DATA LAKE**

1. **Data Lake**
2. **Data Lake là gì**

 **Data Lake** là một kho lưu trữ tập trung, có khả năng chứa lượng lớn dữ liệu từ nhiều nguồn khác nhau, cả có cấu trúc (structured), bán cấu trúc (semi-structured) và không cấu trúc (unstructured).

 Dữ liệu trong Data Lake thường được lưu ở **dạng gốc (raw format)** cho đến khi có nhu cầu xử lý hoặc phân tích.

1. **Tại sao lại sử dụng Data Lake**

 **Linh hoạt**: Lưu trữ mọi loại dữ liệu — logs, video, hình ảnh, văn bản...

 **Chi phí thấp**: Thường dùng hệ thống lưu trữ rẻ như Hadoop, Amazon S3...

 **Dễ mở rộng**: Khả năng scale theo chiều ngang (horizontal scaling).

 **Hỗ trợ phân tích nâng cao**: Machine learning, real-time analytics nhờ dữ liệu đầy đủ và chi tiết.

 **Tích hợp với hệ thống BI, AI/ML** linh hoạt hơn so với data warehouse truyền thống.

1. **Bottoms-up Approach**

Đây là cách xây dựng hệ thống dữ liệu ngược với top-down:

* Thu thập dữ liệu từ các nguồn ngay lập tức vào Data Lake.
* Sau đó mới xây dựng các tầng xử lý, phân tích và báo cáo.

=> Phù hợp với tổ chức cần nhanh chóng thu thập dữ liệu, linh hoạt thay đổi quy trình.

1. **Thiết kế Data Lake**

**Những lưu ý khi thiết kế:**

* Xác định rõ nguồn dữ liệu hiện tại và tương lai.
* Hiểu rõ kích thước, loại, tốc độ dữ liệu.
* Data lake không ép buộc phải có cấu trúc cụ thể lúc ingest dữ liệu → linh hoạt.
* Tuy nhiên cần tổ chức các lớp (layer) để tránh hỗn loạn (data swamp).

**Các lớp (layer) trong Data Lake:**

1. Raw data layer (Bronze layer)
   * Lưu trữ dữ liệu nguyên bản, chưa qua xử lý.
   * Dữ liệu không có cấu trúc, chưa được làm sạch.
   * Định dạng phổ biến: CSV, JSON, Parquet.
2. Conformed layer
   * Chuyển đổi dữ liệu về cùng một định dạng (thường là Parquet).
   * Còn gọi là base/standardized layer.
3. Cleansed data layer (Silver layer)
   * Làm sạch, chuẩn hóa dữ liệu về encoding, cấu trúc, loại dữ liệu.
   * Tạo thành các dataset có thể sử dụng ngay.
   * Có thể xử lý việc tích hợp từ nhiều nguồn.
4. Presentation data layer (Gold layer)
   * Có logic xử lý rõ ràng để phục vụ trực tiếp người dùng cuối, báo cáo, dashboard, ML, AI.
   * Dữ liệu sẵn sàng tiêu thụ.
5. Sandbox data layer
   * Vùng chơi (playground) cho các nhà khoa học dữ liệu, phân tích thử nghiệm.
   * Không bắt buộc phải có, tùy chọn thêm.

**Multiple data lakes**

* Có thể tồn tại nhiều data lake trên toàn cầu cho các khu vực khác nhau → kết nối và phân phối dữ liệu tốt hơn.

**So sánh Data Lake và Data Warehouse:**

| **Tiêu chí** | **Data Warehouse** | **Data Lake** |
| --- | --- | --- |
| Loại dữ liệu | Có cấu trúc | Mọi loại dữ liệu |
| Lược đồ (Schema) | Schema trước (schema-on-write) | Schema sau (schema-on-read) |
| Tốc độ truy vấn | Nhanh với dữ liệu đã xử lý | Thường chậm hơn |
| Chi phí | Cao | Rẻ hơn |
| Phân tích nâng cao | Giới hạn | Hỗ trợ tốt (ML, AI, exploration) |
| Người dùng chính | Business users | Data engineers, data scientists |

1. **Data storage solutions**

| **Thành phần** | **Vai trò trong lưu trữ dữ liệu** | **Mối liên hệ với Storage Solutions** |
| --- | --- | --- |
| ODS | Lưu dữ liệu giao dịch mới nhất (gần real-time), dùng để vận hành hằng ngày. | Có thể lưu trên storage nhanh như SSD, Azure SQL, hoặc dịch vụ OLTP. |
| Data Hub | Trung tâm chia sẻ dữ liệu giữa các hệ thống. Dữ liệu thường chuẩn hóa, có thể thời gian thực hoặc batch. | Lưu trữ trên dịch vụ linh hoạt như S3, GCS, Kafka, hoặc HDFS. |
| Data Mart | Lưu dữ liệu được tổng hợp theo chủ đề để phân tích nhanh cho từng bộ phận. | Dữ liệu được lấy từ Data Warehouse/Data Lake, lưu trên nền tảng tối ưu truy vấn như Redshift, BigQuery, Snowflake. |

1. **Data processes**

a. Master Data Management (MDM)

* Quản lý dữ liệu chủ: khách hàng, sản phẩm, nhà cung cấp…
* Đảm bảo dữ liệu nhất quán, chính xác trên toàn tổ chức.

b. Data Virtualization & Data Federation

* Data virtualization: Truy cập và truy vấn nhiều nguồn dữ liệu như thể chúng là một.
* Data federation: Tập hợp dữ liệu từ các nguồn khác nhau thành một view duy nhất mà không cần sao chép.

c. Data Catalogs

* Giống như thư viện mô tả dữ liệu trong Data Lake.
* Bao gồm: tên dữ liệu, định nghĩa, mô tả, định dạng, ai sở hữu, dữ liệu dùng ở đâu, chất lượng ra sao...
* Ví dụ công cụ: AWS Glue Data Catalog, Apache Atlas.

1. **Approaches to Design**

Một số hướng tiếp cận thiết kế Data Lake:

* Zone-based architecture: Raw zone, trusted zone, refined zone.
* Lambda architecture: Kết hợp xử lý batch và real-time.
* Lakehouse: Kết hợp Data Lake và Data Warehouse (ví dụ: Delta Lake, Apache Iceberg).

**LAKEHOUSE**

**1. Định nghĩa tổng quan**

Lakehouse là một kiến trúc dữ liệu kết hợp ưu điểm của Data Warehouse (DW) và Data Lake, nhằm mang lại:

* Quản lý dữ liệu có cấu trúc như DW,
* Tính linh hoạt và mở rộng như Data Lake,
* Tối ưu chi phí lưu trữ, hiệu năng xử lý,
* Khả năng xử lý BI lẫn ML/AI trên cùng một nền tảng.

**🧱 2. Kiến trúc tổng thể**

Lakehouse thường bao gồm 4 tầng chính:

**🔹 2.1 Ingestion Layer (Tầng thu thập dữ liệu)**

Nguồn: RDBMS, NoSQL, API, Kafka, file system, IoT...

Công cụ: Apache NiFi, Kafka Connect, Airbyte, Fivetran.

Data Format: JSON, Avro, Parquet, ORC, Delta Lake format...

**🔹 2.2 Storage Layer (Tầng lưu trữ)**

Nơi lưu dữ liệu thô hoặc đã qua xử lý sơ bộ.

Dùng file format có khả năng ghi đè/lịch sử như Delta Lake, Apache Hudi, Iceberg.

**Ưu điểm:**

Tách biệt compute/storage,

Versioning, ACID Transaction (nếu dùng Delta/Hudi/Iceberg),

Tối ưu truy vấn bằng file indexing & partitioning.

**🔹 2.3 Processing Layer (Xử lý dữ liệu)**

Batch: Spark, Flink, DBT.

Streaming: Structured Streaming (Spark), Kafka Streams, Flink.

Hỗ trợ cả ETL và ELT (vì compute tách khỏi storage).

**🔹 2.4 Consumption Layer (Tầng truy cập)**

Cho BI/analytics: Power BI, Superset, Tableau.

Cho ML: Databricks ML, Vertex AI, SageMaker.

SQL Engine: Trino, Presto, Dremio, Apache Spark SQL.

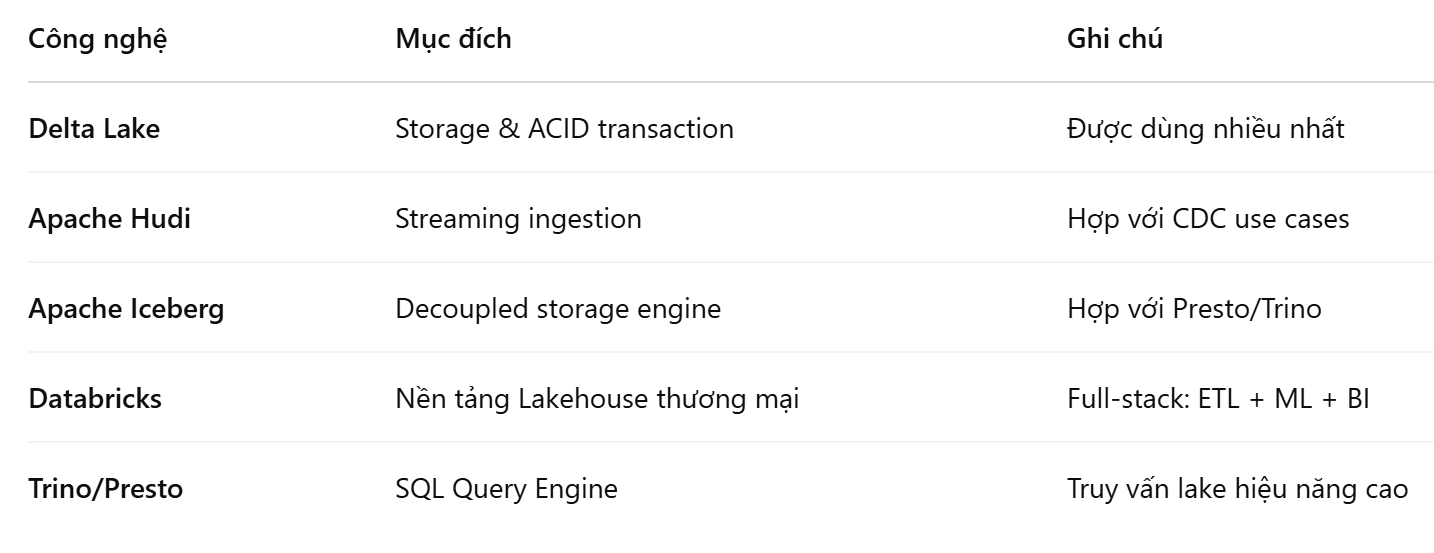
**🔐 3. Ưu điểm vượt trội**



**⚠️ 4. Thách thức kỹ thuật**

* Schema Evolution:
* Phải đồng bộ schema khi dữ liệu thay đổi cấu trúc.
* Cần kiểm soát chặt chẽ ở tầng ingestion.
* Data Governance:
* Quản lý lineage, auditing, security, compliance.
* Giải pháp: Unity Catalog (Databricks), AWS Lake Formation.
* Performance Tuning:
* Partitioning, Z-ordering, caching, file compaction rất quan trọng.
* Phải kiểm soát kích thước file (small files = bad perf).
* Transaction Management:
* Không phải tất cả storage engine đều hỗ trợ ACID.
* Phải chọn engine phù hợp (Delta Lake → Databricks/Spark, Iceberg → Trino/Snowflake).

**🔧 5. Các công nghệ nổi bật**

****

**🛠️ 6. Mô hình triển khai thực tế**

****

**7. Tương lai và xu hướng**

* Lakehouse as a Service: Snowflake, Databricks, BigQuery đang mở rộng theo mô hình này.
* Open Table Formats: Sự cạnh tranh giữa Delta vs Iceberg vs Hudi đang mở ra kỷ nguyên "table format war".
* Real-time Lakehouse: Streaming-first Lakehouse là bước tiếp theo (Flink, Hudi).
* AI Native: Tích hợp MLOps, model tracking (MLflow), feature store ngay trong Lakehouse.