# **Lý thuyết**

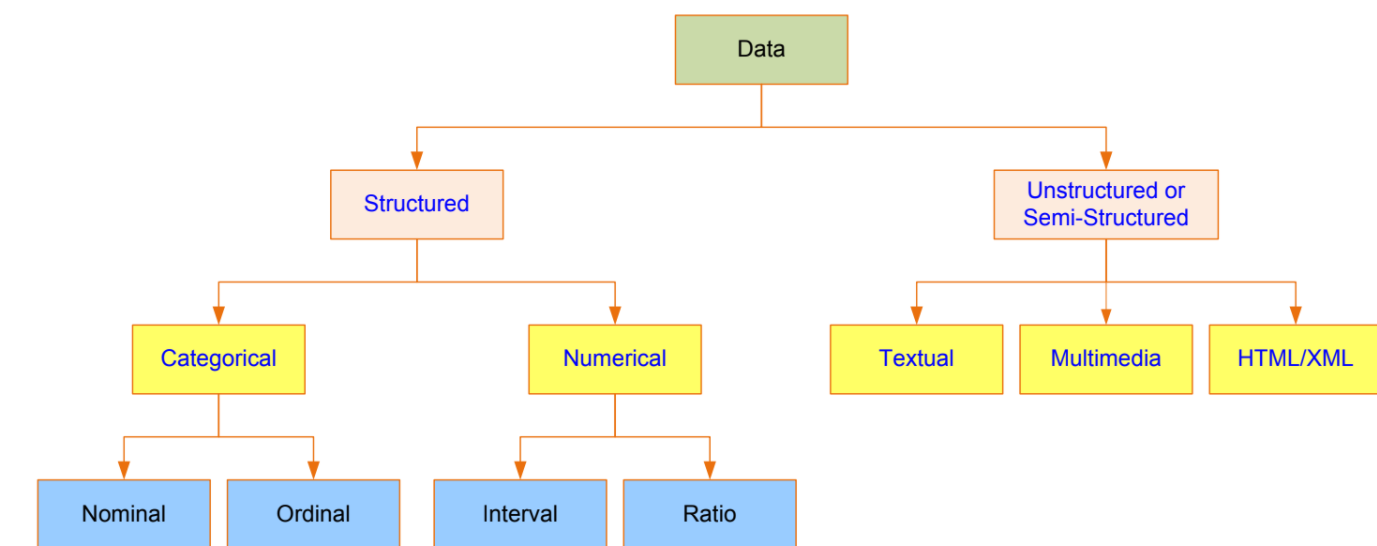
**Câu 1: Data Integration là gì? Mục tiêu của nó và kể tên các công cụ mà em biết.**

**Tích hợp dữ liệu** ([Data Integration](https://gocnhinso.com/redirect-to/?url=https%3A%2F%2Fvi.wikipedia.org%2Fwiki%2FT%25C3%25ADch_h%25E1%25BB%25A3p_d%25E1%25BB%25AF_li%25E1%25BB%2587u)) là công việc đảm bảo hoạt động trao đổi dữ liệu/thông tin giữa các chương trình/hệ thống với nhau để cùng (xử lý) hướng tới mục đích cụ thể nào đó.

Mục tiêu: bao gồm việc kết hợp dữ liệu không đồng nhất trong các nguồn khác nhau vào một lược đồ duy nhất và có thể truy vấn, cung cấp cho người dùng một cái nhìn thống nhất về chúng.

Công cụ:Oracle Data Integrator 12c, SAP Data Services, Xplenty

## **Câu 2. Trong mining, có nhiều loại dữ liệu, cho ví dụ minh họa về 1 loại dữ liệu nào đó?**



* **Kiểu phân loại(categorical)**
* **Định danh *(Nominal data*)**: là tập các nhãn dùng để mô tả, phân loại các đối tượng.

Ví dụ: tên màu, mã nhân viên.

* **Nhị phân *(Binary data)***: là một trường hợp đặc biệt của kiểu định danh, các dữ liệu thuộc kiểu này chỉ mang một trong hai giá trị.

Ví dụ: kiểu boolean (true, false), giới tính (nam, nữ).

* **Thứ tự *(Ordinary data)***: là tập các phần tử chỉ định một thứ tự được sắp.

Ví dụ: xếp loại (kém, trung bình, khá, giỏi).

##### Kiểu liên tục

* **Số nguyên *(Integer)***: là tập các số nguyên, các phần tử thuộc kiểu này chỉ có thể chịu tác động của các phép toán số học để kết xuất phần tử mới.
* **Khoảng *(Interval data)***: dữ liệu khoảng, là một tập các giá trị mà các phần tử cách đều nhau (thường dùng để làm các thang đo).

Ví dụ: nhiệt độ được đo theo độ C.

* **Tỷ lệ-khoảng *(Ratio-scaled data)***: tương tự kiểu dữ liệu khoảng, điểm khác biệt là các phần tử thuộc kiểu dữ liệu này có thể so sánh như là bội số với nhau. Dữ liệu kiểu tỷ lệ có thể thực hiện các phép nhân, chia.

Ví dụ: Trọng lượng: 10kg là hai lần của 5kg. Sự khác biệt giữa 1 và 2 tương tự như khác biệt giữa 3 và 4.

**Câu 3: Thuật toán cây quyết định/gom cụm/luật kết hợp để làm gì? cho ví dụ áp dụng về thuật toán đó.**

Thuật toán cây quyết định thường được dùng trong việc phân chia lớp cho dữ liệu dựa vào dãy các luật. Thuật toán cây quyết định sử dụng dữ liệu ban đầu thuộc những loại như Binary, Ordinal, Nominal để phân lớp cho một thuộc tính thuộc dạng Binary hoặc Ordinal. Sử dụng bộ luật kết hợp sau khi phân lớp để dự đoán lớp cho thuộc tính cần dự đoán trong tương lai

Ví dụ trong thực tế: Giả sử ta có một dataset bao gồm thói quen mua hàng của nhiều người, và cần bán sản phẩm là giấy vệ sinh, với input là những thứ 1 người mua. Các thuộc tính thể hiện người này có mua sản phẩm A không, thuộc tính cần dự đoán là người này có mua giấy vệ sinh không. Sau khi áp dụng cây quyết định ta có thể dự đoán những món đồ nào sẽ được mua cùng với giấy vệ sinh và có thể điều chỉnh đặt giấy vệ sinh lại gần khu chứa những món đồ đó để tăng doanh số bán ra.

**Câu 4: ETL là gì? Có những giải pháp ETL nào?**

a) ETL chính là thuật ngữ được viết tắt bởi cụm từ tiếng Anh Extract Transform Load.

– **Extracts**: Có nghĩa là quy trình này sẽ thu gom nguồn dữ liệu ở các nguồn khác nhau. Để thu gom, doanh nghiệp cần dùng đến một số ứng dụng để đảm nhiệm việc làm nào đó, như nguồn dữ liệu thuộc về người dùng từ hệ thống database trong website, từ hệ thống CMR cũng như gom lại dữ liệu.

– **Transforms**: chuyển đổi nguồn dữ liệu mang mục đích rõ ràng, chính là chuyển từ dữ liệu nghiệp vụ sang dữ liệu đã được phân tích, cũng như tối ưu phân tích dữ liệu. Việc chuyển đổi này còn vì mục đích làm sạch dữ liệu.

– **Load**: sau quá trình dữ liệu được chuyển đổi thì tất cả sẽ được lưu trữ ở một một nơi mới, nơi này được gọi là Kho dữ liệu Data Warehouse.

b) Các giải pháp ETL

- **Enterprise Software ETL:** phần mềm ETL thương mại . Một số công ty phần mềm bán và hỗ trợ các sản phẩm phần mềm ETL thương mại. Chúng đã tồn tại lâu nhất và có xu hướng trưởng thành nhất trong việc áp dụng và chức năng. Tất cả các sản phẩm này đều cung cấp giao diện đồ họa để thiết kế và thi công các đường ống ETL. Tất cả chúng đều kết nối với hầu hết các cơ sở dữ liệu quan hệ. Một số có hỗ trợ các nguồn dữ liệu không quan hệ như JSON và XML. Một số ít hỗ trợ các nguồn phát trực tuyến sự kiện như Apache Kafka.(Vd: Informatica PowerCenter,Oracle Data Integrator,Microsoft SQL Server Integration Services,...)

- **Open Source ETL:** phần mềm ETL mã nguồn mở cho phép bạn mở rộng hoặc nâng cấp. Các công cụ này khác nhau đáng kể về chất lượng, tích hợp, tính dễ sử dụng, áp dụng và tính sẵn có của hỗ trợ. Giống như các công cụ ETL dành cho doanh nghiệp, nhiều công cụ ETL mã nguồn mở này cung cấp giao diện đồ họa để thiết kế và thực hiện các đường ống.(Vd: Pentaho Data Integration,Hadoop,...)

- **Custom ETL:** sử dụng các ngôn ngữ lập trình mục đích chung để viết các công cụ ETL của riêng họ . Cách tiếp cận này có tính linh hoạt cao nhất, nhưng cũng đòi hỏi nhiều nỗ lực nhất. Cách tiếp cận này cũng yêu cầu người dùng thực hiện bảo trì của riêng họ, xây dựng tài liệu của riêng họ, kiểm tra và thực hiện phát triển liên tục. Người dùng ETL tùy chỉnh thường gặp khó khăn khi tìm kiếm sự trợ giúp từ những người bên ngoài nhóm của họ.(VD: SQL, Java, Python,...)

- **Cloud Services:** Amazon AWS, Google Cloud Platform và Microsoft Azure cung cấp các khả năng ETL của riêng họ dưới dạng dịch vụ đám mây.Các dịch vụ này có lợi thế là cung cấp tích hợp chặt chẽ với các dịch vụ đám mây khác, tính đàn hồi và định giá dựa trên mục đích sử dụng.(VD: AWS EMR,Google Cloud Dataflow,Azure Data Factory,...)

**Câu 5: Có 3 loại dữ kiện fact, đó là những dữ kiện nào? Cho ví dụ.**

Có 3 loại dữ kiện fact, bao gồm

* Additive: Là dữ kiện có thể được tổng hợp từ tất cả các dim table liên kết với bảng fact, dữ kiện loại này thường là tính tổng, khi cộng dữ kiện này trên các chiều khác nhau đều cho ra kết quả mang ý nghĩa

Ví dụ: tổng số lượng hàng hóa theo cửa hàng và khu vực, ta có thể thông qua 2 chiều cửa hàng và khu vực để tính tổng số hàng hóa

* Semi-additive: Là dữ kiện có thể được tổng hợp từ một số dim table thay vì toàn bộ. Đối với 1 số chiều dữ kiện này không thể tổng hợp được

Ví dụ: số dư tài khoản ngân hàng không thể tính tổng theo chiều thời gian, nhưng có thể tính tổng theo chiều ngân hàng (nhiều tài khoản khác nhau)

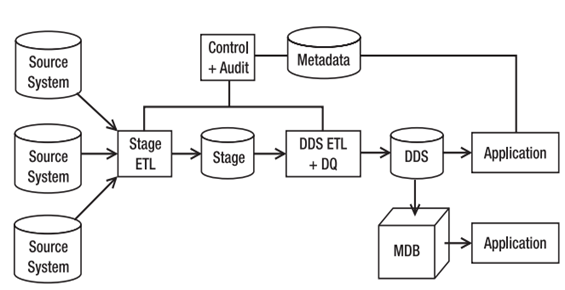
* Non-additive: Là dữ kiện không tính tổng được theo bất kì chiều nào

Ví dụ: tương quan phần trăm giữa doanh thu của 1 cửa hàng so với tiền vốn. Ta không thể tính phần trăm tương quan thông qua tính tổng

**Câu 6. Kiến trúc Single DDS,NDS+DDS,ODS+DDS có ưu khuyết điểm gì?**

**Kiến trúc Single DDS**

Chỉ bao gồm một vùng xử lý (Stage), kho dữ liệu đầu cuối (DDS), Control + Audit là quản lý quá trình của ETL và lưu trữ nhật ký của các kết quả thực thi, metadata là chứa các mô tả, cấu trúc dữ liệu và các tiến trình trong DW.



Kiến trúc DDS đơn là một trong những dạng kiến trúc đơn giản nhất của kho dữ liệu. Kiến trúc này có thành phần chính là một kho dữ liệu trung tâm.

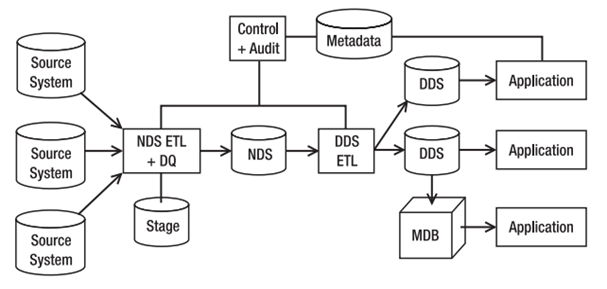
Dữ liệu từ nhiều nguồn được nạp vào vùng xử lí (Stage) thông qua một gói ETL (Extract – Transform - Load). Gói ETL này sẽ rút trích dữ liệu từ nhiều nguồn khác nhau, thực hiện một số phép biến đổi dữ liệu đơn giản và sau đó dữ liệu được đưa vào trong vùng xử lí (Stage). Dữ liệu trong vùng xử lí sau khi được xử lí sơ bộ sẽ được biến đổi thông qua một gói ETL khác để đưa vào kho dữ liệu đầu cuối(DDS). Quá trình biến đổi này bao gồm nhiều công đoạn từ việc làm sạch, chuẩn hoá dữ liệu đến việc quản lí chất lượng và lịch sử thay đổi của dữ liệu. Kho dữ liệu đầu cuối chứa những dữ liệu đã được biến đổi, chuẩn hoá, và lưu trữ dưới dạng mô hình đa chiều, sẵn sàng phục vụ cho các ứng dụng.

**Ưu điểm của kiến trúc DDS đơn:** là kiến trúc đơn giản, truy vấn nhanh và thuận lợi cho việc xây dựng những kho dữ liệu nhỏ.

**Nhược điểm của kiến trúc DDS đơn**: không hỗ trợ việc tạo ra nhiều kho dữ liệu phục vụ cho nhiều mục đích khác nhau dựa trên dữ liệu sẵn có và không tái sử dụng được gói ETL đã làm vì quy trình ETL được thực hiện độc lập.

#### Kiến trúc NDS+DDS

Tương tự như kiến trúc DDS đơn, nhưng có thêm một vùng chứa dữ liệu trung gian là vùng chứa dữ liệu chuẩn hoá NDS. Kiến trúc bao gồm vùng xử lí (Stage), vùng dữ liệu chuẩn hoá (NDS), và kho dữ liệu đầu cuối (DDS).



Dữ liệu sau khi được làm sạch, thay vì đưa thẳng vào DDS, thì nó được lưu trong vùng chứa dữ liệu trung gian. Vùng chứa dữ liệu trung gian đóng vai trò như là một cơ sở dữ liệu tập trung, đã được chuẩn hoá, bao gồm cả dữ liệu lịch sử. Việc nạp vào kho dữ liệu đầu cuối sẽ không cần qua công đoạn làm sạch và quản lí chất lượng dữ liệu nữa.

**Ưu điểm** là chứa dữ liệu thời gian và sẵn sàng cho việc nạp vào nhiều kho dữ liệu đầu cuối và tái sử dụng các gói ETL.

**Nhược điểm** là kiến trúc phức tạp, tốn thêm không gian lưu trữ và thời gian thực hiện một chu kì nạp dữ liệu lâu hơn so với kiến trúc DDS đơn

#### Kiến trúc ODS+DDS

#### Tương tự như kiến trúc NDS+DDS nhưng sử dụng vùng dữ liệu hoạt động thay cho vùng dữ liệu chuẩn hoá.

#### 

#### Hình 2. 5. Kiến trúc kho dữ liệu dạng ODS+DDS

#### NDS được thay bằng ODS (dạng chuẩn 3NF hoặc cao hơn)

#### ODS chỉ chứa dữ liệu hiện tại, không chứa dữ liệu lịch sử, có thể được cập nhật.

#### ODS có thể được truy cập bởi người dùng/ứng dụng cuối

#### DDS là kho dữ liệu chính chứa tất cả dữ liệu lịch sử và hiện tại được dùng khi chỉ có một DDS.

#### Vùng dữ liệu hoạt động này cũng là một cơ sở dữ liệu dạng chuẩn hoá cao. Tuy nhiên, nó không lưu dữ liệu lịch sử. Vùng dữ liệu hoạt động có cấu trúc nghiêng về dạng cơ sở dữ liệu phục vụ giao tác (OLTP) nhiều hơn. Nó đóng vai trò như là một cơ sở dữ liệu tập trung mà ở đó, ứng dụng đầu cuối cho phép khai thác trên nó.

#### Có thể thấy những ưu điểm và nhược điểm của nó so với kiến trúc NDS+DDS như sau:

#### Ưu điểm: Lưu trữ dữ liệu tập trung đã được làm sạch và tận dụng làm cơ sở dữ liệu tập trung phục vụ giao tác cho ứng dụng đầu cuối.

#### Nhược điểm là không chứa dữ liệu lịch sử và không tái sử dụng được các gói ETL.

**Câu 7. Các lược đồ DDS (sao, bông tuyết, chòm sao) có ưu khuyết điểm gì?**

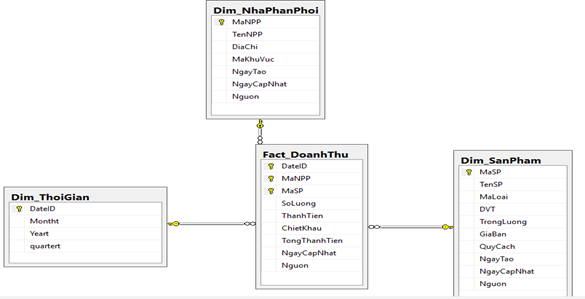
#### Lược đồ hình sao

Lược đồ hình sao là lược đồ diễn tả các dữ liệu theo nhiều chiều gồm có một bảng fact (hay còn gọi là bảng sự kiện) được bao quanh bởi các bảng dimension (hay còn gọi là bảng chiều). Dữ liệu ở đây thì không được chuẩn hóa. Lược đồ hình sao có lợi ích là tách các dữ liệu thành từng lớp dữ liệu nhỏ theo bảng chiều nhằm tăng tốc độ xử lý và dễ dàng phân tích được dữ liệu.

Các thành phần của lược đồ hình sao: bảng sự kiện (Fact), bảng chiều (Dimension viết tắt Dim) và đường kết nối giữa bảng Fact và bảng Dim. Bảng Fact thường là dữ liệu số, có tính toán và bảng Fact có khóa ngoại nhằm để liên kết các bảng chiều. Ví dụ: tính tổng sản phẩm bán ra trong một ngày hay số sản phẩm còn lại trong kho,... Bảng Dim chứa thông tin mô tả các nghiệp vụ thông thường là dữ liệu text và các mô tả. Bảng Dim chỉ có một khóa chính và liên kết với bảng Fact.Ví dụ bảng loại hàng hóa hay khách hàng cung cấp các thông tin về loại hàng hay khách hàng.

**Ưu điểm** là dễ dàng truy vấn, tốc độ truy vấn cao, phù hợp với cách nhìn nhận dữ liệu người dùng và được thiết kế theo hướng nghiệp vụ.

**Nhược điểm** là dữ liệu nguồn cần phải được phi chuẩn hóa, bảo đảm ràng buộc toàn vẹn trong quá trình nạp dữ liệu vào kho và dữ liệu trùng lặp nhiều, khiến cho kích thước kho lớn.



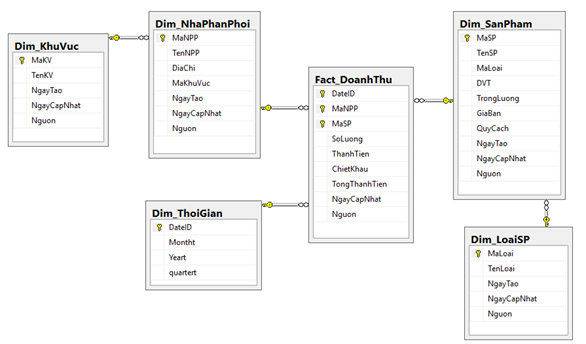
#### Lược đồ hình bông tuyết

Lược đồ hình bông tuyết là giống như lược đồ hình sao nhưng nó được mở rộng từ lược đồ hình sao và được bổ sung thêm các bảng Dim. Các bảng chiều được chuẩn hóa nhằm phân cấp rõ ràng.

Các thành phần của lược đồ hình bông tuyết: bảng sự kiện (Fact), bảng chiều (Dimension viết tắt Dim) và đường kết nối giữa bảng Fact và bảng Dim. Bảng Fact là dữ liệu số, có tính toán và bảng có khóa ngoại nhằm để liên kết các bảng chiều. Bảng Dim chứa thông tin mô tả các nghiệp vụ thông thường là dữ liệu text và các mô tả, bảng Dim chỉ có một khóa chính và liên kết với bảng Fact.

**Ưu điểm** là dữ liệu được nạp từ nguồn vào đích dễ dàng và có dạng chuẩn cao giúp bảo đảm các ràng buộc toàn vẹn của dữ liệu, tránh trùng lắp.

**Nhược điểm** là việc kết các bảng để cho ra kết quả truy vấn mong muốn phức tạp. Bên cạnh đó, còn đòi hỏi người dùng phải hiểu được cấu trúc của kho dữ liệu và tốc độ truy vấn chậm do việc kết các bảng có kích thước lớn.

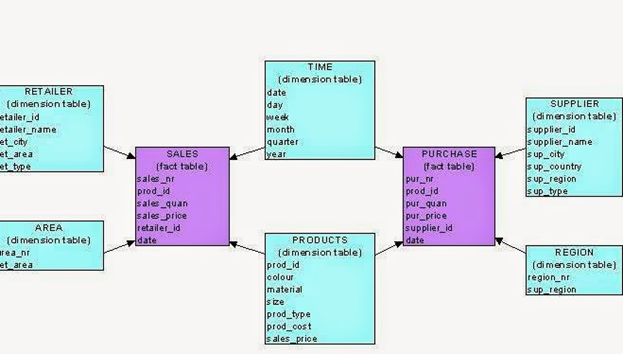


#### Lược đồ chòm sao

Lược đồ chòm sao là sự kết hợp của nhiều Data Mart tức là chứa nhiều bảng Fact và cùng sử dụng chung một số bảng Dim.

Các thành phần của lược đồ chòm sao: bảng sự kiện (Fact), bảng chiều (Dimension viết tắt Dim) và đường kết nối giữa bảng Fact và bảng Dim. Bảng Fact là dữ liệu số, có tính toán và bảng có khóa ngoại nhằm để liên kết các bảng chiều. Bảng Dim chứa thông tin mô tả các nghiệp vụ thông thường là dữ liệu text và các mô tả, bảng Dim chỉ có một khóa chính và liên kết với bảng Fact.

Đặc điểm của lược đồ là có 2 hoặc nhiều bảng fact, tái sử dụng các chiều.



**Câu 8**: **Các đặc trưng của Data WareHouse.**

* Dữ liệu toàn vẹn (Integration)
* Dữ liệu gắn thời gian và có tính lịch sử (Isolation)
* Dữ liệu vận hành ổn định (Non-volatility)
* Dữ liệu cố định (Durable)
* Dữ liệu được hướng theo chủ đề (Subject orientation)

**Lợi ích của data warehouse**

* Sự ra đời của kho dữ liệu nhằm đáp ứng khối lượng dữ liệu cần được xử lý ngày càng nhiều, nhu cầu lưu trữ dữ liệu phát triển khi các hệ thống máy tính trở nên phức tạp hơn. Cụ thể, kho dữ liệu sẽ đem lại các lợi ích sau đây cho các doanh nghiệp:
* Tích hợp dữ liệu lại về một nguồn, cùng một định dạng, giải quyết tính phân mảnh và rời rạc của dữ liệu nhằm đáp ứng mọi yêu cầu về thông tin của người sử dụng
* Tiết kiệm thời gian và hiệu quả khi tìm thấy dữ liệu cần thiết
* Giúp việc sử dụng dữ liệu của doanh nghiệp trở nên hiệu quả hơn bằng tính năng xử lý và phân tích dữ liệu
* Hỗ trợ người sử dụng đưa ra những quyết định hợp lý, nhanh chóng, năng suất cao hơn, thu được lợi nhuận cao hơn…
* Giúp cho tổ chức, xác định, quản lý và điều hành những dự án, những nghiệp vụ một cách hiệu quả và chính xác.
* Giúp tăng cường khối lượng dữ liệu cần được tổng hợp, lưu trữ và xử lý lên đáng kể
* Hiện kho dữ liệu đã được ứng dụng rộng rãi trong các doanh nghiệp thuộc nhiều lĩnh vực khác nhau, cụ thể ta có thể kể đến:
* Hàng không: Xây dựng hệ thống quản lý các công việc hàng không như quản lý tuyến bay, lịch bay, phân công phi hành đoàn, quản lý các chương trình ưu đãi dành cho khách hàng…
* Ngân hàng: quản lý dòng tiền, quản lý các quỹ đầu tư, cho vay, thời hạn thanh toán…
* Y – Dược: quản lý thông tin bệnh nhân, tình trạng bệnh án, phác đồ điều trị, đơn thuốc được kê cho họ, thời gian điều trị bệnh…
* Bảo hiểm: phân tích xu hướng của khách hàng và để theo dõi sự thay đổi của thị trường một cách nhanh chóng
* Bán lẻ: quản lý mặt hàng nhập vào – bán ra, quản lý chuỗi phân phối, xác định mô hình mua hàng của khách hàng, các chương trình khuyến mãi cũng như các chính sách giá
* Giáo dục: quản lý thông tin học sinh – giáo viên – công nhân viên của trường, quản lý quá trình học tập, giáo án, bài giảng, kết quả học tập của học sinh… Là nền tảng để xây dựng các phần mềm dạy học hiện nay.
* Quản lý quan hệ khách hàng: tổng hợp, xử lý, phân tích và giúp dự đoán insight của khách hàng và xu hướng thị trường trong tương lai

## **Câu 9. Các loại chiều thay đổi chậm. Cho ví dụ từng loại**

*Nạp các chiều thay đổi chậm (Slowly Changing Dimension – SCD):* Là chiều có thuộc tính thay đổi giá trị rất chậm theo thời gian vì một lí do nào đó. Có 3 loại:

* SCD loại 1 (ghi đè): đây là loại chiều không cần lưu lại lịch sử thay đổi. Chỉ việc ghi đè lên bản ghi cũ.
* SCD loại 2 (dữ liệu lịch sử hết hiệu lực): đây là loại chiều cần lưu lại lịch sử. Thay vì ghi đè lên chiều cũ, người ta tạo ra một dòng mới với cùng khoá tự nhiên nhưng khác khoá đại diện. Lúc đó, chỉ cần thay đổi tham chiếu từ bảng fact.
* SCD loại 3 (dữ liệu lịch sử còn hiệu lực): đây là trường hợp các giá trị lịch sử vẫn còn hiệu lực sử dụng đồng thời với các giá trị mới. Thay vì tạo thêm một dòng mới trong bảng chiều, người ta tạo thêm các cột mới để lưu vết.

Thông thường, người ta tránh sử dụng loại 2 vì nó làm thay đổi cấu trúc của hệ thống. Hơn nữa, việc xác định tính hiệu lực của dữ liệu thường được quy định trong nghiệp vụ và được lưu như là một thuộc tính bình thường của chiều đó.

**Câu 10: Tại sao cần sử dụng metadata ?**

* Sử dụng metadate trong DW dùng để ghi lại những thông tin của các sources cũng như trạng thái cập nhật dữ liệu cuar sources vào DW. Metadate có thể được gọi là “Data của data”.
* Metadate có thể chứa tất cả các loại thông tin về dữ liệu DW như:
* Nguồn cho bất kỳ dữ liệu trích xuất nào.
* Sử dụng dữ liệu DW đó.
* Bất kỳ loại dữ liệu nào và các giá trị của nó.
* Đặc điểm của dữ liệu.
* Logic chuyển đổi cho dữ liệu được trích xuất.
* Bảng DW và các thuộc tính của chúng.

+ Đối tượng DW

+ Nhãn thời gian

**Câu 11: Operational databases là gì?**

* ODS là cơ sở dữ liệu được sử dụng để quản lý và lưu trữ dữ liệu trong thời gian thực. ODS là nguồn cho một kho dữ liệu. Các phần tử trong cơ sở dữ liệu hoạt động có thể được thêm và xóa nhanh chóng. Các cơ sở dữ liệu này có thể dựa trên SQL hoặc NoSQL, trong đó cơ sở dữ liệu này hướng tới các hoạt động thời gian thực.

**Câu 12: Database và DW?**

* **Giống nhau**: điều dùng để lưu trữ dữ liệu cho hệ thống của tổ chức doanh nghiệp. Dữ liệu của Database và DW có thể chứa dữ liệu hiện tại
* **Khác nhau**:

|  | DW | Database |
| --- | --- | --- |
|  |  |  |
| Định nghĩa | - DW là một hệ thống thông tin lưu trữ dữ liệu lịch sử và dữ liệu giao hoán từ một hoặc nhiều nguồn. Nó được thiết kế để phân tích, báo cáo, tích hợp dữ liệu giao dịch từ các nguồn khác nhau.  - DW giúp giảm bớt quá trình phân tích và báo cáo của một tổ chức. Nó cũng là một phiên bản duy nhất cho tổ chức cho quá trình ra quyết định và dự báo. | -Cơ sở dữ liệu là một tập hợp các dữ liệu liên quan đại diện cho một số yếu tố của thế giới thực. Nó được thiết kế để xây dựng và cung cấp dữ liệu cho một nhiệm vụ cụ thể. |
| Mục đích | được thiết kế để phân tích | được thiết kế để lưu lại các dữ liệu |
| Phương thức | OLTP | OLAP |
| Bảng và phép kết (join) | Bảng và phép nối rất đơn giản trong kho dữ liệu vì chúng không được chuẩn hóa. | Các bảng và các phép nối của cơ sở dữ liệu rất phức tạp vì chúng được chuẩn hóa. |
| Giới hạn lưu trữ | Dữ liệu từ nhiều nguồn khác nhau liên quan trong hệ thông doanh nghiệp | Thông thường sẽ là dữ liệu từ một application |
| Data Type | Dữ liệu lịch sữ và dữ liệu hiện tại sẽ không được cập nhật | Dữ liệu có thể được cập nhật |
| Storage of data | Data Warehouse sử dụng cách tiếp cận theo chiều và chuẩn hóa cho cấu trúc dữ liệu. Ví dụ: Lược đồ ngôi sao và bông tuyết. | Tiếp cận đa số theo hướng relation database store |
| Query Type | Các truy vấn phức tạp được sử dụng cho mục đích phân tích. | Các truy vấn giao dịch đơn giản được sử dụng. |

**Câu 13. Định nghĩa DW của 2 ông Ralph Kimball và Bill Inmon Characteristics of data warehouse in W.H. Inmon definition are: Subject-oriented, integrated, time-varying, non-volatile**

* **Định nghĩa Data warehouse**
* **Định nghĩa 1:**
* Là hệ thống rút trích(retrieves) và hợp nhất(consolidates) dữ liệu theo định kỳ từ source vào một nơi chứa dữ liệu được chuẩn hóa (normalized) hoặc là đa chiều(dimensional)
* Lịch sử dữ liệu được giữ qua nhiều năm và được truy vấn trong các hoạt động BI hoặc các hoạt động phân tích khác. Sẽ được định kỳ update vào KDL, chứ không có update liên tục vào mọi thời điểm
* **Theo Vincent Rainardi**
* **Định nghĩa khác**
* a DW is a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management’s decision-making process - **theo W.H. Inmon ⇒ Normalized data store**

**=> *Data Warehouse là một tập hợp dữ liệu mang tính hướng chủ đề, tích hợp, biến đổi theo thời gian và ổn định, nhằm mục đích hỗ trợ ra quyết định của nhà quản lý.***

* a data warehouse is a system that extracts, cleans, conforms, and delivers source data into a dimensional data store and then supports and implements querying and analysis for the purpose of decision making - **Theo Ralph Kimball ⇒ Demensional data store**

**=> kho dữ liệu là một hệ thống trích xuất, làm sạch, tuân thủ và cung cấp dữ liệu nguồn vào kho dữ liệu chiều, sau đó hỗ trợ và triển khai truy vấn và phân tích nhằm mục đích ra quyết định**

**Câu 14. Enterprise Data Warehouse là gì?**

Enterprise data warehouse (EDW) là kho dữ liệu quan hệ chứa dữ liệu kinh doanh của công ty (bao gồm cả thông tin về khách hàng của công ty). EDW cho phép phân tích dữ liệu, có thể cung cấp thông tin chi tiết hữu ích. Giống như tất cả các Data Warehouse, EDW thu thập và tổng hợp dữ liệu từ nhiều nguồn, hoạt động như một kho lưu trữ cho hầu hết hoặc tất cả dữ liệu tổ chức để tạo điều kiện truy cập và phân tích rộng rãi.

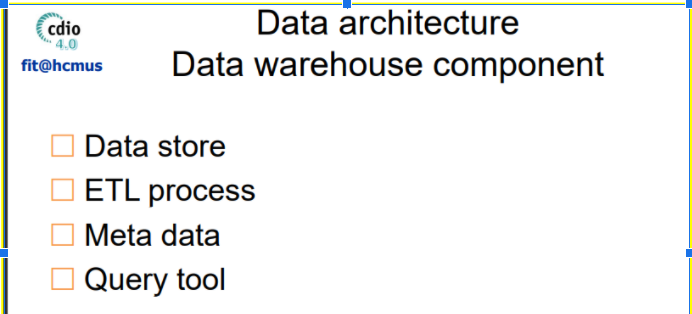
**Ưu điểm của EDW**

* Cung cấp một góc nhìn toàn diện về doanh nghiệp hỗ trợ người phân tích dự đoán và ra quyết định thông minh hơn.
* Có khả năng cung cấp đầy đủ thông tin hiện tại và lịch sử của doanh nghiệp và sẵn sàng cho việc khai thác, sử dụng cho việc hỗ trợ ra quyết định chiến lược.
* Tiết kiệm thời gian trong quá trình truy xuất dữ liệu phục vụ trong quá trình phân tích -> cung cấp các báo cáo quản trị doanh nghiệp nhanh chóng.
* Nâng cao lợi thế cạnh tranh của doanh nghiệp
* Có khả năng mở rộng, không giới hạn.
* Đảm bảo dữ liệu an toàn

**Khuyết điểm của EDW**

* Tốn nhiều thời gian trong quá trình thiết kế,tích hợp làm sạch và tải dữ liệu vào kho.
* Làm tăng nhanh nguồn dữ liệu.
* Tốn nhiều không gian lưu trữ.
* Trong quá trình tích hợp và đồng nhất dữ liệu từ nhiều nguồn có khả năng mất mát dữ liệu có giá trị.
* Tốn nhiều chi phí lưu trữ, bảo trì và vận hành EDW

**Câu 15. The four components of data architecture là gì ?**

****

**Câu 16. So Sánh Data mart/ Single data mart**

**Ưu điểm:**

* Tích hợp dữ liệu lại về một nguồn, cùng một định dạng, giải quyết tính phân mảnh và rời rạc của dữ liệu nhằm đáp ứng mọi yêu cầu về thông tin của người sử dụng
* Tiết kiệm thời gian và hiệu quả khi tìm thấy dữ liệu cần thiết
* Giúp việc sử dụng dữ liệu của doanh nghiệp trở nên hiệu quả hơn bằng tính năng xử lý và phân tích dữ liệu
* Hỗ trợ người sử dụng đưa ra những quyết định hợp lý, nhanh chóng, năng suất cao hơn, thu được lợi nhuận cao hơn…
* Giúp cho tổ chức, xác định, quản lý và điều hành những dự án, những nghiệp vụ một cách hiệu quả và chính xác.
* Giúp tăng cường khối lượng dữ liệu cần được tổng hợp, lưu trữ và xử lý lên đáng kể

**Khuyết điểm:**

* Đánh giá thấp tài nguyên tải dữ liệu. Thông thường, chúng ta không ước tính được thời gian cần thiết để truy xuất, làm sạch và tải dữ liệu lên kho. ...
* Các vấn đề tiềm ẩn trong hệ thống nguồn
* Chi phí bảo trì có khi lớn hơn lợi ích
* Đồng nhất dữ liệu.

**Ưu điểm của single data mart:**

* Data mart chứa tập con của dữ liệu của toàn tổ chức, tăng tính tối ưu hóa cho 1 nhóm người dùng cụ thể trong một tổ chức
* Chi phí để xây dựng ít hơn data warehouse, vì thường data warehouse sẽ có chi phí xây dựng cao hơn
* Cho phép truy cập dữ liệu nhanh hơn
* Vì được thiết kế chuyên biệt cho nhu cầu nhóm người cụ thể nên sẽ đẩy nhanh các tiến trình trong kinh doanh hơn
* Việc triển khai data mart nhanh hơn nhiều so với data warehouse vì chúng ta chỉ cần làm việc với 1 tập dữ liệu con thay vì toàn bộ dữ liệu
* Chứa những lịch sử dữ liệu cho phép phân tích và dự đoán xu hướng của dữ liệu 1 cách dễ dàng
* Data Mart giúp tăng cường thời gian phản hồi của người dùng do giảm khối lượng dữ liệu.
* Dữ liệu rất dễ dàng để truy cập thường xuyên

**Khuyết điểm của Data mart:**

* Mất chi phí cao để xây dựng.
* Các doanh nghiệp tạo ra quá nhiều data marts khác nhau và không liên quan . Nó có thể trở thành một trở ngại lớn để duy trì. Data Mart không thể cung cấp phân tích dữ liệu toàn công ty vì bộ dữ liệu của họ bị hạn chế
* Có thể gặp phải sự phân mảnh.

**Câu 19.1. Tại sao phải lấy CET ở bước 2, không phải là B4?**

Các bước :

1. Lấy thông tin LSET được lưu trong metadata

2. Lấy CET: thời gian khởi động ETL package

3. Rút trích dữ liệu:

select \* from order\_header where (created >= LSET and

created < CET) or (last\_updated >= LSET and last\_update

< CET).

4. Cập nhật LSET = CET

**Trả lời:** Cả 2 câu đều bắt nguồn chung từ một vấn đề là độ trễ và thời gian phải load dữ liệu từ mỗi nguồn có thể dẫn đến sót dữ liệu.

**Câu 19.2: Cần set CET ở bước 2 không phải bước 4 vì như giải thích ở câu 2**, CET được sử dụng trong bước rút trích (B3), nên cần phải lấy ở bước 2. Giả sử trong select không sử dụng CET mà dùng hàm getNow() lấy thời gian hiện tại thì rõ ràng thời điểm rút trích với thời điểm set CET ở bước 4 là khác nhau(do quá trình rút data cần mất một khoảng thời gian). Do đó để đảm bảo không bị lệch thời gian và không sót dữ liệu thì cần có CET và phải set trước khi rút trích ở bước 3.

**Câu 20. Tại sao cần chặn trên CET?**

Nếu không có CET, tức là LSET được set sau khi select data, sẽ có trường hợp nguồn một load xong trước nguồn 2, nguồn 2 tiếp tục rút trích thì có thêm/update data vào nguồn 1, đây sẽ là dữ liệu bị sót/sai do data mới không được update trong khi LSET được set sau khi rút trích(sau thời gian update data). Nếu có chặn trên CET thì data được thêm vào sẽ được rút trích trong lần ETL sau.

