

**Chương 3 - Mô hình hóa chiều dữ liệu**

Giảng viên: Hồ Thị Hoàng Vy  
2013, Jan 27

1

## 0. Các khái niệm

- Là 1 kỹ thuật thiết kế mức luận lý, phù hợp cho các ứng dụng BI và kho dữ liệu.
- Phục vụ các báo cáo, truy vấn và phân tích trong BI.
- Khái niệm chính của mô hình hóa chiều: dữ kiện (facts), chiều (dimension) và thuộc tính (attribute)
- Chiều có thể có các phân cấp khác nhau và tính hạt (granularity) quy định mức chi tiết của chiều
- Dữ kiện, chiều, thuộc tính có thể được tổ chức theo nhiều cách khác nhau → gọi là lược đồ (schema). Việc lựa chọn lược đồ tùy thuộc vào các biến như: loại báo cáo cần đáp ứng, loại công cụ BI được dùng

2

## 0. Khái niệm

- Fact: phép đo lường hoạt động kinh doanh (business activity) như: 1 sự kiện kinh doanh hay 1 giao dịch kinh doanh, thường là số học.
  - VD: doanh số bán hàng, mức độ tồn kho, chi tiêu... → đo lường số học gồm: số lượng hàng hoá, khoảng tiền (dollars), phần trăm, tỷ lệ...
  - Fact có thể được tính tổng hợp hoặc tính toán. Ví dụ: có thể cộng tổng doanh thu hay tính toán lợi nhuận từ một tập các giao dịch bán hàng
  - Fact được chuẩn hoá và ít thông tin dư thừa, số bộ dữ liệu trong bảng fact sẽ rất lớn. (90% dữ liệu trong mô hình chiều dữ kiện chủ yếu nằm ở bảng fact)

3

## Nội dung

- Giới thiệu**
- Xác định yêu cầu nghiệp vụ**
  - Thiết kế bảng dữ liệu**
  - Thiết kế bảng chiều**
- Mô hình dữ liệu**
- Phân cấp chiều dữ liệu**

4

## 1. Giới thiệu

NDS

subscription sales

product sales

subscriber profitability

Supplier

Performance

Nhận diện nhu cầu khai thác kho dữ liệu

5

## Giới thiệu

- Các hệ thống OLTP thường sử dụng **cấu trúc dữ liệu dạng chuẩn**:
  - Có thể nhất quán hàng trâm giao tác cá nhân
  - Giảm thiểu rủi ro mất dữ liệu, sai dữ liệu
  - Yêu cầu kiến thức về SQL
- ➔ **không phù hợp** đối với những người làm kinh doanh (người đưa ra quyết định chiến lược)
  - Không cần phải học để viết code, không nên tốn thời gian cho các hoạt động code
  - Tốn thời gian viết các report
  - Truy vấn hàng ngàn dòng thông tin để báo cáo → chậm, nặng
- ➔ **mô hình hóa dữ liệu dạng chiều** phù hợp với các xử lý ra quyết định
  - Biểu diễn bởi 2 loại bảng: FACT & DIMENSION (*chương 2*)

6

## 2. Xác định nhu cầu

### a. Tìm hiểu hoạt động kinh doanh

- Do hệ thống mở rộng trên nhiều quốc gia và vận hành trên các hệ thống khác nhau nên khó khăn trong việc tổng hợp/ phân hồi thông tin bán hàng toàn cầu.
- Cần phân tích **doanh số** theo sản phẩm và khu vực địa lý
- Cần đánh giá **khả năng cung ứng** dựa vào phân phối thời gian xúc tiến, chỉ đạo, số lượng và chất lượng đơn hàng, lịch sử giao dịch và trả nợ...
- Công ty muốn **chọn lựa những khách hàng tiềm năng** dựa vào thông tin địa lý, lịch sử đặt hàng, quyền thông tin, để gửi mail quảng cáo, ghi nhận phản hồi của họ (mở mail, vào website...)
- .....

7

## 2. Xác định nhu cầu

### b. Định nghĩa yêu cầu

#### - Yêu cầu chức năng (Functional requirements):

- định nghĩa những gì hệ thống làm
- Chứa các đặc trưng mà DW cần có

#### - Yêu cầu phi chức năng (nonfunctional requirements)

- Những hướng dẫn và các ràng buộc hệ thống
- Tính Bảo mật, tính sẵn sàng...
- Hiệu suất

8

## 2. Xác định nhu cầu

### • Yêu cầu chức năng

- Ví dụ 1:
  - Cần phân tích doanh số bán hàng (khi khách hàng mua 1 sản phẩm chứ không phải đặt 1 đơn hàng) qua thời gian theo khu vực địa lý, thông tin nhân khẩu, khu vực chứa và bán hàng
  - Cần biết được doanh số, chi phí, lợi tức...
- Ví dụ 2:
  - Cần phân tích việc bán theo đơn đặt hàng (khi khách hàng đặt đơn hàng) theo vùng miền địa lý, khu vực chứa và bán hàng
  - Cần biết lợi nhuận, chi phí, sự chênh lệch giữa giá bán và giá vốn, đánh giá hàng ngày, đều đặn mỗi tháng

9

## 2. Xác định nhu cầu

### • Yêu cầu phi chức năng

- Ví dụ:
  - Các ứng dụng front-end dựa trên nền web, có thể truy cập bất cứ đâu trong mạng công ty
  - Kho dữ liệu phải được sao lưu đều đặn mỗi tháng
  - Công ty sử dụng MS server để xây dựng kho dữ liệu từ đầu tới cuối, bao gồm công cụ ETL, reporting, OLAP.
  - Kho dữ liệu phải được thiết kế linh động để có thể dễ dàng cải tiến và tương thích với các thay đổi xảy ra trong hệ thống giao tác, cần cập nhật mới dữ liệu vào kho, có thể thêm các cube/report/ràng buộc dữ liệu, hoặc chỉnh sửa cái hiện thời
  - .....

10

## 2. Xác định nhu cầu

### • Nhu cầu phân tích → sự kiện (event)

- Sự kiện: là 1 hoạt động lặp đi lặp lại theo 1 khoảng thời gian (vài phút, vài giây, vài ngày...)
- Ví dụ:
  - Phân tích doanh số bán hàng → Sự kiện khách hàng mua hàng → hóa đơn
  - Phân tích kết quả học tập → Sự kiện học → Kết quả

11

## 2. Xác định nhu cầu

### • Nhu cầu phân tích → sự kiện (event)

bối cảnh + giá trị đo lường

Dimension tables      Fact tables

```

graph TD
    subgraph Dimension_tables [Dimension tables]
        Dim1[Dim 1]
        Dim3[Dim 3]
    end
    subgraph Fact_tables [Fact tables]
        FACT[FACT]
        Dim2[Dim 2]
    end
    Dim1 --> FACT
    Dim3 --> FACT
    FACT --> Dim2

```

12

## 2. Xác định nhu cầu

- Ví dụ 2.1:**
  - Phân tích doanh số bán hàng theo 1 số bối cảnh: khách hàng, sản phẩm, thời gian....

13

## 2. Xác định nhu cầu

- Ví dụ 2.2:** Phân tích kết quả học tập → Sự kiện học → Kết quả

14

## 3. Mô hình dữ liệu

- a. Mô hình sao**
- b. Mô hình bông tuyết**
- c. Mô hình chòm sao**

15

## 3.1 Mô hình sao

- Đặc điểm**
  - Không có subdimension
  - Đơn giản
  - Quá trình ETL load dữ liệu vào DDS dễ dàng
  - Hiệu suất truy vấn tốt do ít phép kết
  - Dư thừa dữ liệu cao và tốn nhiều không gian đĩa

16

## 3.2 Mô hình bông tuyết

- Đặc điểm**
  - Là sự mở rộng từ mô hình sao
  - Có subdimension → giảm tối thiểu dư thừa dữ liệu
  - Sử dụng cho các ứng dụng phân tích tốt hơn các mô hình còn lại

17

## 3.3 Mô hình chòm sao

- Đặc điểm**
  - Có 2 hoặc nhiều bảng fact, tái sử dụng các chiều
  - Có khả năng mô hình hóa nhiều sự kiện kinh doanh

18

## 4. Thiết kế bảng dữ kiện

- Bảng Dữ kiện gồm 2 loại cột: **khoá (keys)** và **độ đo (measures)**
- Mỗi liên hệ giữa bảng dữ kiện và bảng chiều là: một – nhiều

19

## 4.1 Khoá của bảng dữ kiện

- Khoá của bảng dữ kiện gồm: một nhóm các khoá ngoại trở tới các khoá chính của các bảng chiều có liên đới với bảng dữ kiện để có thể phân tích nghiệp vụ.
- Khoá chính của bảng dữ kiện là 1 khoá nhiều thành phần bao gồm sự kết hợp của các khoá ngoại để xác định 1 dòng duy nhất trong bảng dữ kiện

20

## 4.1 Khoá của bảng dữ kiện

- Ví dụ 4.1.1:** ngữ cảnh khi khách hàng mua sản phẩm, có thể dùng mô hình sau để lưu giữ thông tin về quy trình nghiệp vụ:



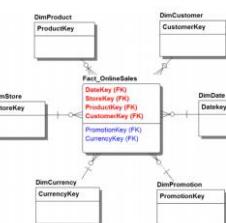
21

## 4.1 Khoá của bảng dữ kiện

### Ví dụ 4.1.2: Các cột khoá trong mô hình bên:

- DateKey—Date and time purchase was made
- StoreKey—Online store in which purchase was made
- ProductKey—Product that was bought
- CustomerKey—Customer who made purchase
- PromotionKey—Sales promotion
- CurrencyKey—Currency used to make purchase

➔ **Khoá chính:** (DateKey, StoreKey, ProductKey, CustomerKey)



22

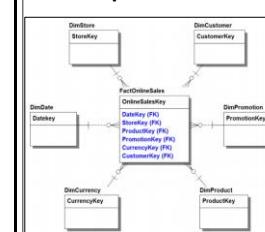
## 4.1 Khoá của bảng dữ kiện

- Nếu sự kết hợp của tổ hợp khoá ngoại không thể xác định duy nhất 1 dòng trong bảng dữ kiện, có 2 phương pháp:
  - Sử dụng khoá đại diện (surrogate key)
    - Được phát sinh bởi hệ thống dữ liệu, kiểu integer, giá trị tự tăng (identity). Tuy nhiên (xem VD 4.1.3)
  - Sử dụng chiều thoát hoá/suy biến (degenerative dimension) (xem VD 4.1.4)

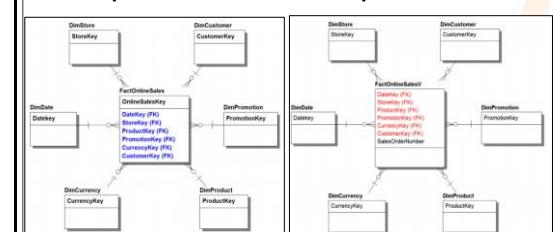
23

## 4.1 Khoá của bảng dữ kiện

### Ví dụ 4.1.3



### Ví dụ 4.1.4



24

## 4.2 Độ đo của bảng dữ kiện

- Là giá trị đo lường (measure) thực sự của 1 hoạt động kinh doanh như: tiền lời từ việc bán hàng, hay số lượng đặt hàng
- Mỗi sự đo lường đều có 1 tính chất hạt (grain) – chỉ cấp độ chi tiết trong việc đo lường 1 sự kiện (đơn vị đo lường, tiền tệ, số dư tài khoản cuối ngày,...)
  - Tính chất hạt của tiền tệ có thể là: dollar amount, hoặc chi tiết hơn: xu (cents)
  - Tính chất này được quyết định bởi nguồn dữ liệu

25

## 4.2 Độ đo của bảng dữ kiện

- Tính hạt (GRANULARITY)**

- Dùng để chỉ mức chi tiết được lưu trữ trong bảng fact.
- Tính hạt càng cao → giới hạn khả năng lấy thông tin mức chi tiết
- Ví dụ:** nếu chiều thời gian biểu diễn bởi mỗi năm → mỗi sự kiện bán hàng lưu giữ cho mỗi năm → không thể xem thông tin chi tiết theo mỗi tháng, mỗi tuần...
- Tính hạt thấp → mở rộng kích thước của kho dữ liệu so với nhu cầu
- Ví dụ:** mỗi record thời gian biểu diễn cho 1 giờ → sẽ có 1 sự kiện bán hàng được lưu trữ theo mỗi giờ của ngày???

26

## 4.2 Độ đo của bảng dữ kiện

- Ví dụ 4.2.1**

- several measures of an internet sale:* chỉ 1 khách hàng mua 1 sản phẩm tại 1 thời điểm cụ thể. Tất cả các độ đo này liên quan tới sự kiện bán hàng mà bảng dữ kiện biểu diễn và có mức độ tính chất chi tiết liên quan đến sự kiện đó.

Fact_OnlineSales	
PK	DateKey
PK	StoreKey
PK	ProductKey
PK	CustomerKey
CURRENTKEY	PromotionKey
	SaleOrderNumber
	SaleOrderLineNumber
	SaleQuantity
	SaleAmount
	RefundAmount
	DiscountQuantity
	DiscountAmount
	TotalCost
UNITS	UnitCost
UNITS	UnitPrice
DW	ETLLoadID
DW	LastUpdateDate
DW	UpdateDate

27

## 4.3 Chiều & phân cấp chiều

- Là 1 thực thể thiết lập nền ngữ cảnh nghiệp vụ cho các độ đo của bảng dữ kiện.
- Các chiều định nghĩa: ai, cái gì, ở đâu và tại sao của mô hình chiều và nhóm các thuộc tính tương tự vào 1 loại hoặc 1 chủ đề.
  - Các chiều như: sản phẩm, khách hàng, nhân viên, thời gian...
  - Tạo chiều giúp cho bảng dữ kiện chỉ lưu các thuộc tính ở 1 nơi thay vì nhân bản dữ thừa qua các dòng của bảng dữ kiện
  - Các bảng chiều được chuẩn hoá

28

## 4.3.1 Khoá của chiều

- Khoá để đảm bảo mỗi dòng trong bảng chiều là duy nhất:

DimProduct		DimGeography	
ProductKey		GeographyKey	
ProductAlternateKey		PostalCode	
WeightUnitMeasureCode		City	
SizeUnitMeasureCode		StateProvinceCode	
EnglishProductName		StateProvinceName	
StandardCost		CountryRegionCode	
FinishedGoodsFlag			
Color			
SafetyStockLevel			
ReorderPoint			
ListPrice			
Size			
SizeRange			

29

## 4.3.1 Khoá của chiều

- Khoá đại diện (surrogate key):**

- Dùng làm khoá chính, được phát sinh bởi hệ thống dữ liệu, là số nguyên, tự tăng và giá trị của nó không mang ngữ nghĩa.
- Lý do:**
  - khi thu thập dữ liệu chiều từ nhiều nguồn, thường có các khoá không nhất quán, không tương thích được dung qua các hệ thống này
  - Khoá chính của các hệ thống nguồn thường thay đổi theo thời gian, các hệ thống nguồn theo thời gian có thể được thay thế
  - ...
  - Khoá chính của nguồn chỉ nên làm khoá thay thế trong bảng chiều, còn gọi là khoá tự nhiên của nguồn dữ liệu
  - Nếu có nhiều nguồn, cần 1 thuộc tính xác định nguồn

30

### 4.3.1 Khoá của chiều

#### Khoá đại diện (surrogate key):

- Ví dụ: bảng Dim\_Customers:
  - CustomerSK: Khoá chính của bảng chiều
  - CustomerNK: khoá tự nhiên trong bảng chiều và là khoá chính của nguồn
  - SOR\_NK: nguồn của record này
  - khoá thay thế nhiều phần trong bảng chiều này chính là (SOR\_NK và CustomerNK)

Dim_Customers	
PK	CustomerSK
SOR_NK	CustomerNK
U1	Title
	FirstName
	MiddleName
	LastName

31

### 4.3.2 Phân cấp chiều

- Chiều thường được phân cấp.
- Mỗi kết hợp của các phân cấp: nhiều – một:
  - Sản phẩm – loại sản phẩm
  - Nhóm địa lý: khu vực bán hàng
  - Cấu trúc tổ chức: tiếp thị/bán hàng
  - Thời gian: năm-tháng-quý-tuần-ngày-giờ-phút

DimGeography	
GeographyKey	PostalCode
	City
	StateProvinceCode
	StateProvinceName
	CountryRegionCode

33

### 4.3.2 Phân cấp chiều (dimension hierachy)

- Phi chuẩn hoá chiều → chiều có các phân cấp ~ gom nhóm, cấu trúc hoá**

- Ví dụ:** phân cấp chiều thời gian, cửa hàng, sản phẩm



35

### 4.3.2 Phân cấp chiều (dimension hierachy)

- Ví dụ:** thực hiện báo cáo dữ liệu sử dụng các chiều khác nhau, tại các cấp khác nhau

Năm	Vùng miền	Doanh thu
1996	Asia	1000
	Europe	50000
	America	20000
1997	Asia	1500

➔ Kết quả bảng trên thực hiện trên các cấp nào, chiều nào?

36

### 4.3.2 Phân cấp chiều (dimension hierachy)

- Khoá của Fact liên kết khoá của Dimensions**

- Thực hiện báo cáo doanh thu theo tháng**

- Tổng quát từ cấp Ngày → Tháng
- Gom nhóm dữ liệu theo Tháng của Năm (*idThang*)



37

### 4.3.2 Phân cấp chiều (dimension hierachy)

Thống kê dữ liệu theo các phân cấp chi tiết hơn trong mỗi chiều

Năm	Quý	Vùng miền	Doanh thu
1996	Q1	Asia	200
	Q2	Asia	200
	Q3	Asia	250
1997	Q4	Asia	350
	Q1	Europe	10000
----	----	-----	-----



38

### 4.3.2 Phân cấp chiều (dimension hierachy)

- Cấp chi tiết nhất

The diagram illustrates the dimension hierarchy for three dimensions:

- Thời gian (Time):** Year (Năm), Quarter (Quý), Month (Tháng), Day (Ngày).
- Cửa hàng (Store):** All (Tất cả), Region (Vùng), Country (Quốc gia), City (Thành phố).
- Sản phẩm (Product):** All (Tất cả), Product - Department (Sản phẩm - phòng ban), Product - Type (Sản phẩm – Loại), Product (Sản phẩm).

Days are grouped under Months, which are grouped under Years. Cities are grouped under Regions, which are grouped under Countries. Products are grouped under Departments, which are grouped under Product Types.

39

### 4.3.2 Phân cấp chiều (dimension hierachy)

- Trong các bảng chiều: 1 thuộc tính (attribute/column) có thể là 1 tập con gồm các thuộc tính khác
  - Giá trị của 1 thuộc tính được gom nhóm bởi các thuộc tính khác
  - Thuộc tính được dùng để gom nhóm ở cấp tổng quát hơn thuộc tính bị gom nhóm
  - Ví dụ:** 1 năm gồm 4 quý; 1 quý = 3 tháng → thứ tự cấp tổng quát nhất đến chi tiết nhất: Năm → Quý → Tháng
- Rollup - drill down:**
  - Phân tích dữ liệu lên mức tổng quát hơn/ xuống mức chi tiết hơn.
  - Ví dụ: nếu biết được doanh thu của mỗi tháng trong quý → doanh thu của quý đó

40

### 4. Phân cấp chiều

- Cấp thời gian:**
  - Quan trọng
  - Cần ghi nhận tất cả các cấp thời gian cần thiết cho việc phân tích và phù hợp nội dung phân tích
  - Ví dụ:** phân tích số liệu tài chính của 1 doanh nghiệp

The diagram shows a hierarchical structure of time periods:

- Fiscal Year
- Year
- Quarter
- Fiscal Period
- Month
- Week
- Date
- Fiscal Week

42

### Modeling the calendar

- Calendar trong business:
  - Gregorian calendar (chuẩn)
  - fiscal calendar (kế toán, quản lý tài chính)
  - billing-cycle calendars
  - factory calendars

43

### Modeling the calendar

The diagram shows the relationship between Date, Day, Month, and Year:

- Date:** Date, Date Identifier (FK), Month Identifier (FK), Year Identifier (FK), Day Sequence Number.
- Day:** Day Identifier, Day Name, Day Short Name, Day Workday Indicator.
- Month:** Month Identifier, Month Name, Month Short Name.
- Year:** Year Identifier, Year Number.

Dashed lines indicate foreign key relationships. The text "Gregorian calendar" is written in red.

**Figure 6.1** Calendar in the business model.

44

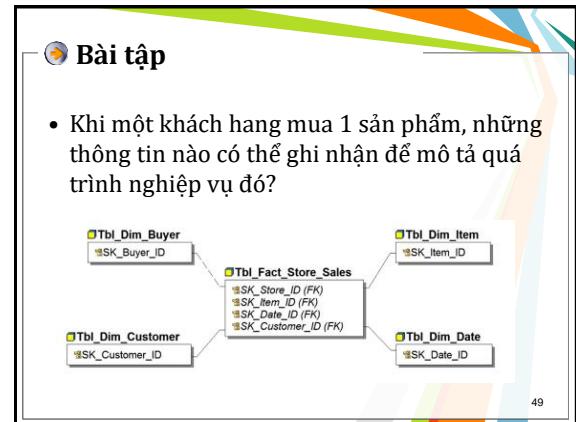
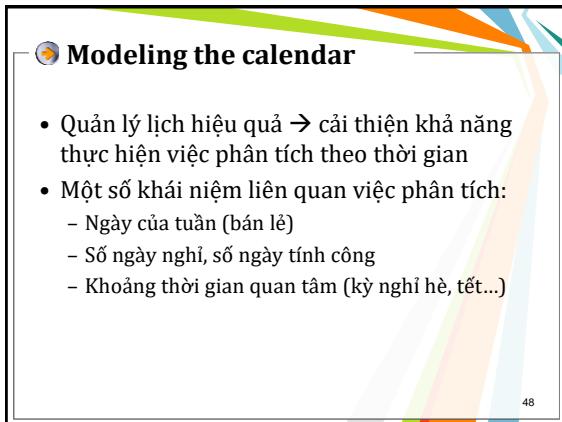
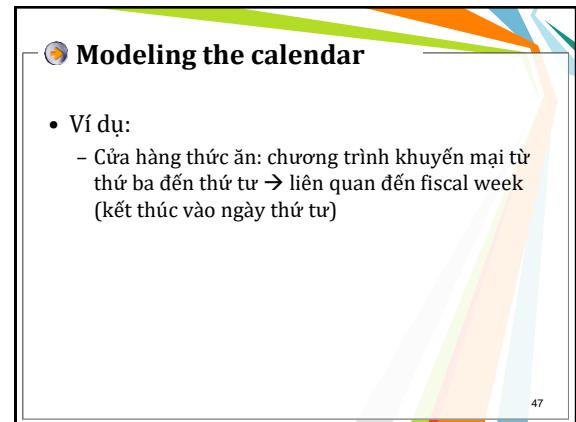
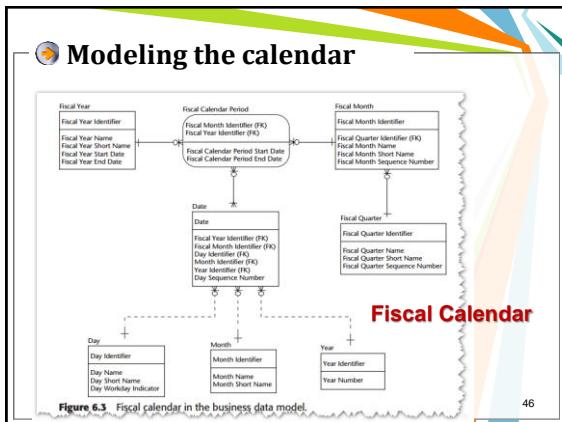
### Modeling the calendar

The diagram shows the relationship between Fiscal Year, First Quarter, Second Quarter, Third Quarter, Fourth Quarter, and Months (July, August, September):

- Fiscal Year:** First Quarter, Second Quarter, Third Quarter, Fourth Quarter.
- First Quarter:** July, August, September.
- Second Quarter:** July, August, September.
- Third Quarter:** July, August, September.
- Fourth Quarter:** July, August, September.

A vertical column on the left lists days 1 through 31. The text "Fiscal Calendar" is written in red.

45



## Câu hỏi

- Fact (sự kiện) và dimension (chiều) lưu trữ liệu được chuẩn hoá hay phi chuẩn? Nêu lý do.
- Mức chi tiết của dữ kiện có liên quan đến chiều không?  
Giải thích & cho ví dụ minh họa
- Cho các bảng
  - Chiều 1: Ngày -> tháng -> quý -> năm
  - Đợt khai giảng → mỗi 2 khoá khai giảng → mỗi 4 khoá KG → năm
  - Chuyên đề → nhóm học phần → ngành
  - Dữ kiện: ngày, tháng, quý, năm, đợt KG, mỗi 2 khoá KG, mỗi 4 khoá KG, năm, chuyên đề, nhóm HP, ngành, số lượng SV\_đăng kí, SL\_HV\_datyeucuu

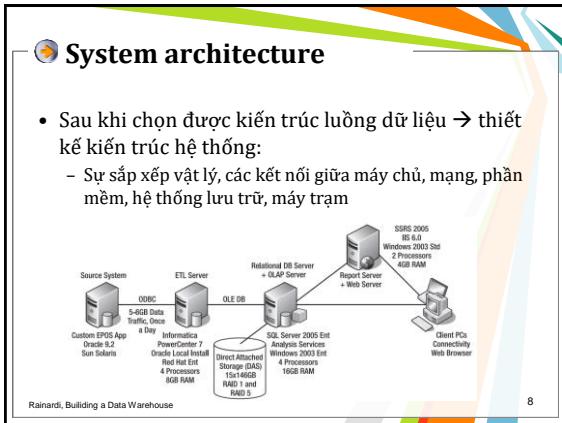
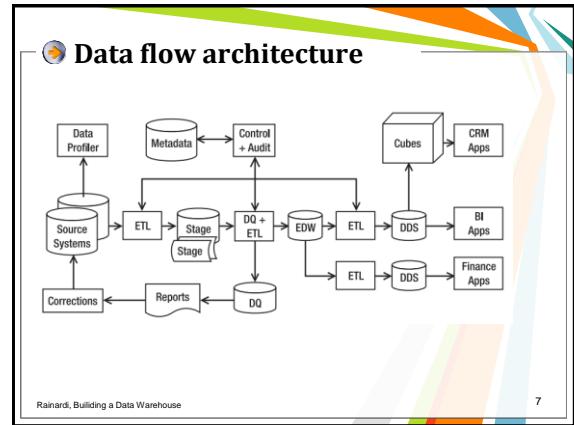
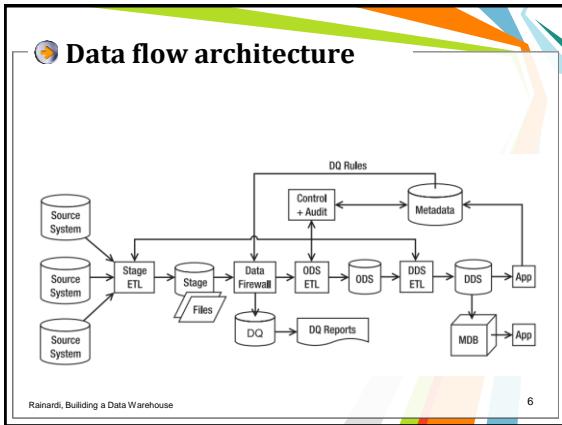
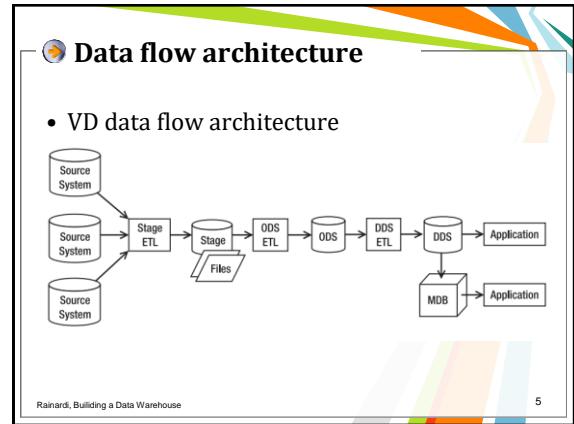
**Câu hỏi:** cho biết mức chi tiết của các dữ kiện. Có thể tính số lượng học viên đăng kí với các mức chi tiết của các chiều khác nhau không

52

## Sửa bài

1. Dữ kiện phải được diễn giải theo tiêu chí (~chiều) gì. Do đó, cấp (mức chi tiết) của dữ kiện phụ thuộc vào mức chi tiết của chiều
  - Ví dụ: doanh số bán hàng của mỗi ngày (chiều thời gian) tại mỗi cửa hàng (chiều địa điểm). Chiều thời gian được ghi nhận ở cấp chi tiết ngày → doanh số ngày
2. Dữ kiện có mức chi tiết nhất : ngày (chiều 1), đợt khai giảng (chiều 2), chuyên đề (chiều 3)
3. Có thể tính được.
  - Ví dụ chiều 1 chọn cấp quý, chiều 2 chọn cấp mỗi 2khoá, chiều 3 chọn chi tiết cấp chuyên đề. Đơn từ cấp tổng quát xuống ta có:
    - Chiều 1 : cấp 2, chiều 2: cấp 3, chiều 3: cấp 3
    - tổng hợp chiều 2 lên 1 cấp → số lượng sv đăng kí 2K
    - Tổng hợp chiều 1 lên 2 cấp từ kết quả trên → số lượng sv đăng kí 2K\_Quy

53



## Thiết kế KDL

- Tìm hiểu nhu cầu:
  - Chi tiết các quy trình, nghiệp vụ, dữ liệu, các vấn đề
  - Thu thập các kinh nghiệm người dùng, thảo luận ngữ nghĩa dữ liệu, giao diện..
  - Liệt kê các nhu cầu chức năng & phi chức năng, hiệu suất và bảo mật

13

## Thiết kế KDL

### Kiến trúc:

- Xác định kiến trúc hệ thống đang dùng
  - Đặc tả chi tiết: server dữ liệu, loại hình mạng, giải pháp lưu trữ,
- Quyết định kiến trúc dữ liệu sẽ dùng
  - Trả lời các câu hỏi:
    - Không gian lưu trữ
    - Khả năng xử lý

14

## Thiết kế KDL

- Thiết kế
  - Kho chứa dữ liệu (data store)
  - Hệ thống ETL
  - Các ứng dụng Front-end
  - Hệ thống đảm bảo chất lượng dữ liệu (data quality system) và meta data
- Cài đặt
- Kiểm chứng
- Triển khai

15

## Thiết kế KDL

### Vận hành

- Quản trị kho và hỗ trợ người dùng
  - Giải quyết lỗi và vấn đề khi sử dụng hệ thống
  - Quản trị người dùng mới
  - Nhu cầu phát sinh: thêm dữ liệu mới, thêm yêu cầu, , tạo report mới, cube mới, thêm tính năng trong ứng dụng front-end...

### Thiết lập hạ tầng

- Mua/cài đặt/cấu hình/kiểm chứng phần cứng, phần mềm

### Quản lý dự án

16

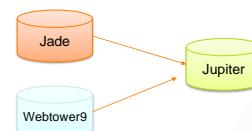
## VẬN DỤNG

- Phân tích case study
  - Đồ án lý thuyết

17

## Hiện trạng

- Công ty Amadeus Entertainment có 3 nguồn dữ liệu:
  - Jade (DB2) – Product Sale Offline
  - Jupiter (SQL server) – Inventory & products
  - Webtower9 (Oracle) – Product sales (online)



18

## Hiện trạng

- Công ty là một nhà bán lẻ chủ yếu: films, audio books, music.
  - Có tám cửa hàng online vận hành ở Mỹ, Đức, Pháp, Anh, Tây Ban Nha, Úc, Nhật và Ấn Độ.
  - Có 96 cửa hàng offline cũng trên những quốc gia này.
  - Khách hàng có thể mua hàng cá nhân với các sản phẩm: bài hát, audio book, phim, hoặc đăng ký gói cho phép download một số sản phẩm trong 1 khoảng thời gian nhất định
  - Công ty có nhiều kênh phân phối: TV, internet, mobile phone...
  - Có nhiều phương thức chi trả: trả hàng năm, trả trước hàng tháng...

19

## Nhu cầu

- Người dùng nghiệp vụ cần phân tích bán hàng (product sales). Cần biết được doanh số, chi phí, lợi nhuận theo một loại tiền tệ (\$)
- Người dùng nghiệp vụ cần phân tích "bán hàng qua hình thức đăng ký" (subscription sales). Cần biết được doanh số, chi phí, lợi nhuận được đánh giá hằng ngày trong khoảng 1 tháng
- Cấp cửa hàng, có thể xem dữ liệu hằng ngày trong một vài tuần trước đó ở mức tổng quát hoặc chi tiết về doanh số, chi phí, lợi nhuận để hiểu nguyên nhân của việc doanh số thấp, lợi nhuận thấp, sản phẩm nào, khách hàng nào... là nguyên nhân của vấn đề

20

## Yêu cầu

- Cấp quản lý toàn cục có thể hiểu được xu hướng chung tổng thể hoặc từng quốc gia. Liệu một cửa hàng hay một quốc gia đang có vấn đề với một sản phẩm nào để có thể tương tác sớm với các cửa hàng nhất có thể
- Hệ thống có thể hiển thị các hình, biểu đồ cho phép in được, có thể xuất ra excel hoặc csv.
- Phục vụ phân tích các chiến dịch chăm sóc khách hàng qua việc xem các do lường của mỗi chiến dịch sau khi gửi cho khách hàng: số lượng thông điệp được gửi của mỗi kênh (mobile message, email, post...), số thông điệp phân phối thành công, thất bại, tỷ lệ mở, tỷ lệ bỏ qua, than phiền, đánh spam, ...

21

## Yêu cầu

- Phục vụ phân đoạn khách hàng cho các chiến dịch chăm sóc khách hàng dựa vào quyền thông tin, đặc trưng địa lý, thông tin khách hàng, sở thích, lịch sử mua hàng...
- Phân tích hiệu suất nhà cung ứng (supplier performance) – trung bình chi phí, giá trị trả về, từ chối, thời gian ngừng hoạt động..

22

## Kiến trúc cài đặt

The diagram illustrates the data warehouse architecture. Key components include:
 

- Source Systems:** Jade Informix AS4, Jupiter AS400 DB2, WebTower9 Win 2000 SQL Server 2000, Storage Area Network (SAN) with 12TB Raw Capacity 85 146GB 15k Disks.
- ETL Servers:** ETL Server (OLE DB), NDS & DDS Database Server, OLAP Server.
- Report Servers:** Report Servers + Web Farm (2-Node Web Farm Win 2003 R2 EE 4 Processors 16GB RAM).
- Analysis Services:** SQL Server 2005 Analysis Services Win 2003 R2 EE 4 Processors 16GB RAM.
- Network Infrastructure:** Gigabit Network, Fibre Channel Switches, Fiber Network.
- Storage:** 200-500 Client PCs.

 Arrows indicate data flow from sources through ETL to report and analysis servers, with network links connecting them.

23

## Kiến trúc luồng dữ liệu

The diagram shows the data flow architecture in iteration 1. The process starts with data from various sources (Jade, Jupiter, Web Tower) feeding into a central NDS ETL + DQ stage. This stage then feeds into a NDS layer, which in turn feeds into a DDS ETL layer. The DDS ETL layer outputs to DDS, which then feeds into Reports. A separate Control + Audit layer monitors the process, and a Metadata layer provides metadata support. A Stage layer is also shown between the NDS and DDS layers.

24

## Phân tích yêu cầu

- Người dùng nghiệp vụ cần phân tích bán hàng (product sales). Cần biết được doanh số, chi phí, lợi nhuận theo một loại tiền tệ (\$)

### Sự kiện:

- Khi 1 khách hàng mua 1 sản phẩm

### Bối cảnh sự kiện:

- Ai: khách hàng
- Ở đâu: cửa hàng, lãnh thổ bán hàng
- Cái gì: sản phẩm
- Khi nào: ngày mua hàng

### Đo lường (dữ kiện): Số lượng, đơn giá, giá trị

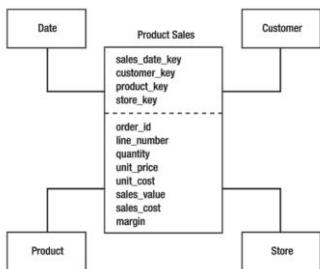
25

## Mô hình hóa

- Đo lường → Fact Table
- Bối cảnh → dimension table
- Mô hình sao liên kết các dimension table và fact table
- Phân cấp dữ liệu

27

## Mô hình hóa



Rainardi, Building a Data Warehouse

28

## Mô hình hóa - Fact table

- Các giá trị có sẵn từ nguồn:
  - Quantity, unit\_price, unit\_cost
- Các giá trị phải tính toán:
  - Sales\_value, sales\_cost, margin
    - $\text{sales\_value} = \text{unit\_price} \times \text{quantity}$
    - $\text{sales\_cost} = \text{unit\_cost} \times \text{quantity}$
    - $\text{Margin} = \text{sales\_value} - \text{sales\_cost}$

### Cấp chi tiết dữ liệu (độ mịn)

- Đơn vị nhỏ nhất xảy ra sự kiện:
  - Một dòng trong fact tương ứng ứng mỗi item được bán

29

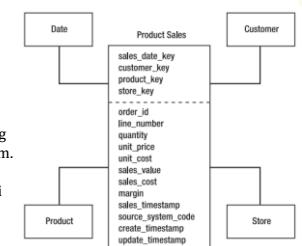
## Mô hình hóa - Fact table

- Nhu cầu phức tạp hơn: discounts, taxes, promotional items?
- Xem lại nguồn dữ liệu
- Luôn xác nhận mô hình dữ liệu với các logic nghiệp vụ cùng hệ thống nguồn để đảm bảo kho dữ liệu phản ánh đúng các điều kiện nghiệp vụ

30

## Mô hình hóa - Fact table

- Chiều thoái hoá
- Nhân thời gian
- Khoá chính của fact?
  - 1 khách hàng A, mua bài hát B, tại cửa hàng C vào ngày D lúc 10am. Sau đó quay lại lúc 7pm để mua cùng bài hát lần nữa?



Rainardi, Building a Data Warehouse

31

## Mô hình hóa - Fact table

- Order\_ID, line\_number
  - WebTower9 or Jade order ID;
  - WebTower9 or Jade order line number
- Source\_system\_code
  - Phân biệt dữ liệu từ nguồn nào (key của nguồn)
- Sales\_timestamp
  - Thời gian khách hàng mua sản phẩm
- Create\_timestamp – update\_timestamp
  - Thời gian dòng dữ liệu được tạo/cập nhật trong DDS

32

## Mô hình hóa - Thiết kế chiều

- Các chiều liên quan sự kiện phân tích:
  - Date dimension
  - Product dimension
  - Customer dimension
  - Store dimension
- Lưu giá trị cũ của chiều
  - Xem chiều thay đổi chậm

34

## Mô hình hóa - Thiết kế chiều

### Date dimension

- Tại sao không lưu time dimension thay vì date dimension

35

## Mô hình hóa - Thiết kế chiều

### Product dimension

- Không yêu cầu lưu giá trị cũ
  - » SCD 1 - Ghi chép giá trị cũ
- Chỉ có 1 nguồn là từ Jupiter
  - » chuyển product từ Jupiter → Product dimension
- » Nếu từ nguồn bảng product được chuẩn hóa → phi chuẩn → join các bảng với nhau

36

## Mô hình hóa - Thiết kế chiều

- Customer dimension
  - Cần lưu trữ giá trị cũ:
    - occupation, household\_income, address1, address2, address3, address4, city, state, zipcode, country, phone\_number, email\_address
    - SCD loại 2
  - Nếu cần lưu nhiều địa chỉ, email, điện thoại
    - Lưu nhiều thuộc tính (star schema)
    - Chuẩn hóa thành các bảng con (snowflake schema)

37

## Mô hình hóa - Thiết kế chiều

### Store dimension:

- Cần lưu giá trị cũ:
  - Hiện tại có 5 region, tương lai có thể tăng thêm
  - Các cửa hàng online hiện chỉ phân bố trong 1 division, tương lai có thể phân bố vật lý như các cửa hàng offline (hiếm xảy ra)
- SCD loại 3 → ko thay đổi nhiều thuộc tính, biết chắc số lượng phiên bản thay đổi
  - Region & prior\_region & prior\_region\_date
  - Division & prior\_division & prior\_division\_date

38

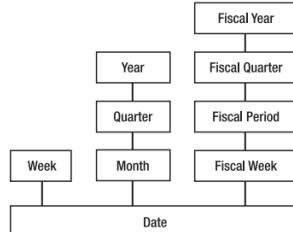
## Mô hình hóa - Phân cấp dữ liệu

- Phục vụ nhu cầu rollup, drilldown dữ liệu
- Trong bảng chiều, đôi khi một thuộc tính là tập con của 1 thuộc tính khác
  - Giá trị của thuộc tính được gom nhóm bởi thuộc tính khác
- Thuộc tính cấp cao: thuộc tính bị dung để gom nhóm
  - Vd: 1 năm có 4 quý, 1 quý gồm 3 tháng → cấp cao nhất là năm, tiếp theo là quý và tháng
  - Nếu biết được doanh thu bán hàng của mỗi tháng trong quý → biết được doanh số của quý đó

39

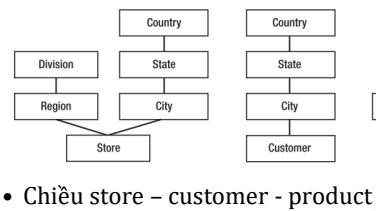
## Mô hình hóa - Phân cấp dữ liệu

- Chiều date



40

## Mô hình hóa - Phân cấp dữ liệu



- Chiều store – customer - product

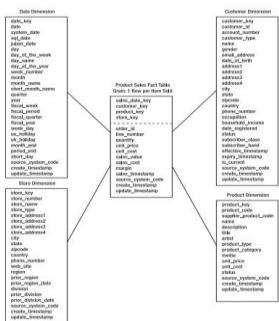
41

## Mô hình hóa

- Sau khi thiết kế xong “product sales” data mart → tiếp tục thiết kế các data mart khác theo cách tương tự:
  - Xác định fact table
  - Thiết lập mức chi tiết (độ min dữ liệu)
  - Xác định dimension table + SCD
  - Xây dựng lược đồ (sao, bông tuyết...)

42

## Mô hình hóa



43

## Ánh xạ dữ liệu nguồn

- Sau khi thiết kế các data mart, cần ánh xạ dữ liệu DDS và nguồn
  - Ánh xạ mỗi column trong fact table và dimension table vào dữ liệu nguồn
- Mục đích: để xác định dữ liệu mỗi column trong DDS sẽ đến từ đâu**
  - Xác định các tính toán, chuyển đổi nếu có → giúp cho ETL
  - 1 dds column có thể đến từ nhiều table ở nguồn

44

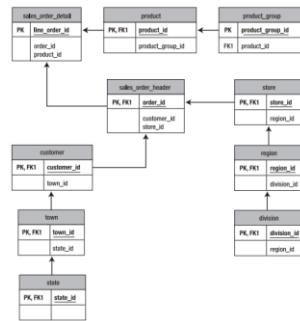
## Thiết kế NDS

- Liệt kê:**
  - Các bảng nguồn
  - Các bảng chuẩn hoá từ fact và dimension table
- Kết hợp hai danh sách, phân chia thành các chủ đề khác nhau:**
  - Thiết lập mối liên hệ giữa các thực thể
  - Thiết kế kho
- Kết nối bảng cha và con (fact → con; dimension → cha trong NDS)
- Chuẩn hoá dữ liệu các bảng từ 3NF trở lên

45

## Thiết kế NDS

- Ví dụ



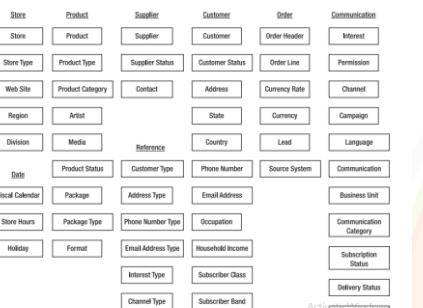
46

## Thiết kế NDS

- Lập bảng mô tả cho các bảng trong NDS**
  - Tên cột, kiểu dữ liệu, mô tả, nguồn, hình thức biến đổi
- Quay lại DDS, định nghĩa lại nguồn của mỗi cột của DDS trong NDS phục vụ cho ETL giữa NDS và DDS

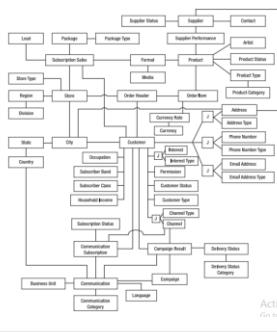
47

## Thiết kế NDS



48

## Thiết kế NDS



49

## Câu hỏi

- Nêu tư tưởng thiết kế vừa rồi (bottom up hay topdown,...)

50

## Ví dụ

- Subscription Sales fact table
  - Bảng Fact có 6 dimension:
    - lead, package, format, store, customer, **date**
  - Bảng Transaction nối với:
    - lead, package, format, store, customer
    - Chiều ngày nên ánh xạ vào NDS như thế nào?**

51

## Gợi ý

Brief introduction to functions used in script to populate Date Dimension

Function	Detail (e.g. for 16-Aug-2013)
1 Select DATEPART(MM, Getdate()) as MonthNumber	Return Integer Number=8 of Month from Current Date
2 Select DATEPART(YY , Getdate()) as YearValue	Return Value of the Year=2013 from Current Date
3 Select DATEPART(QQ , Getdate()) as QuarterValue	Return Value of the Quarter=3 for Current Date
4 Select DATEPART(DW, Getdate()) as DayOfWeekValue	Return integer Value of day=6 (Friday) in Week for Current Date as per US standard
5 Select CONVERT (char(8),Getdate(),112)	Return Key=20130816 Value for current Date
6 Select CONVERT (char(10),Getdate(),103)	Return date = 16/08/2013 in 'dd-MM-yyyy' format: UK, Europe
7 Select CONVERT (char(10),Getdate(),101)	Return date=08/16/2013 in 'MM-dd-yyyy' format: US
8 Select DATEPART(DD , Getdate()) as DayOfMonthValue	Return integer Day=16 Value for Current Date
9 Select DATENAME(DW, Getdate()) AS DayName	Return Name=Friday of the Day for Current Date.
10 Select (DATEPART(WW, Getdate())) AS WeekOfYear	Returns Value of Week in Year=33

53

## ĐỒ ÁN LT

Chọn 1 trong 2

- Thiết kế data mart cho nhu cầu phân đoạn chiến dịch trong quản lý liên hệ khách hàng (CRM)
- Người dùng nghiệp vụ cần phân tích “bán hàng qua hình thức đăng ký” (subscription sales). Cần biết được doanh số, chi phí, lợi nhuận được đánh giá hằng ngày trong khoảng 1 tháng

54

Thank You!

TB: Hồ Thị Hoàng Vy  
2013, Jan 27

55

# Chương 1 – Kho DL Định nghĩa, đặc điểm

Giảng viên: Hồ Thị Hoàng Vy  
2013, Jan 27

1

## Nội dung

- 1. Tình huống dẫn nhập**
- 2. Kho dữ liệu (DW) là gì**
- 3. Các kiến trúc**
- 4. Đặc điểm kho dữ liệu**
- 5. Câu hỏi thảo luận**

3

### 1. Tình huống 1

- Công ty ABC Entertainment là một công ty có **nhiều chi nhánh** tại Mỹ, Đức, Pháp, Anh và hàng chục cửa hàng online & offline hoạt động tại các vùng miền tại các đất nước này.
- **Mỗi chi nhánh có 1 hệ thống vận hành riêng.**

→ **Nhân viên sale manager muốn lập báo cáo doanh thu theo từng quý.**

4

### 1. Tình huống 1 (tt)

Online Transaction Processing (OLTP) system

Anh  
Pháp  
Mỹ  
Đức

Doanh số bán hàng từng sản phẩm Tại tổng công ty trong từng quý

Sales Manager

5

### 1. Tình huống 1 (tt)

Online Transaction Processing (OLTP) system

Anh  
Pháp  
Mỹ  
Đức

Xử lý các tác vụ thông qua các giao tác khai thác CSDL  
Quản lý dữ liệu hiện thời, rất chi tiết  
Không chứa dữ liệu lịch sử hoặc dữ liệu của các tổ chức khác nhau.  
Thùa nhận mô hình thực thể kết hợp (ER) & các ứng dụng CSDL hướng đối tượng  
Đối tượng dùng: nhân viên bán hàng, nhân viên quản trị CSDL, chuyên gia CSDL

6

### 1. Tình huống 1(tt)

(OLTP) system

Data Warehouse (DW)

Báo cáo  
Sales Manager  
Công cụ Truy vấn và phân tích

Cần 1 csdl tổng hợp từ nhiều nguồn  
Rút trích thông tin bán hàng từ csdl tổng hợp

→ OLTP: cung cấp nguồn dữ liệu cho DW

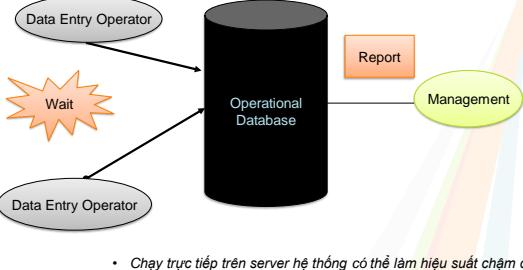
7

## 1. Tình huống 2

- 1 siêu thị bán hàng có một csdl giao dịch rất lớn. Mỗi khi muốn thực hiện các báo cáo, hệ thống OLTP này trở nên chậm và các thao tác trên dữ liệu phải chờ một lúc lâu.

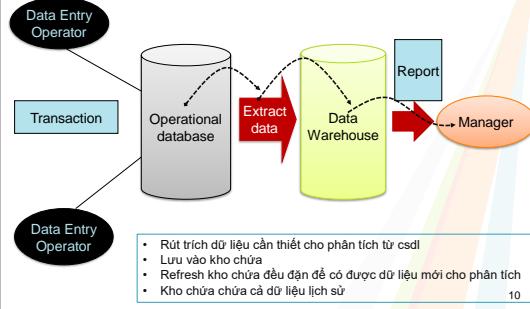
8

## 1. Tình huống 2(tt)



9

## 1. Tình huống 2(tt)



10

## 1. Vấn đề

- Khi các công ty cần lập báo cáo tổng kết, ra quyết định, dự đoán → sử dụng các công cụ IT

### Khó khăn:

- Mất nhiều thời gian
- Không thuận tiện với các vấn đề khám phá dữ liệu
- Các tập dữ liệu lớn, các truy vấn nhiều bảng phức tạp
- Những thay đổi trong report cần phải chỉnh sửa các ứng dụng
- Cần khám phá các thông tin từ dữ liệu hiện tại và lịch sử để đưa ra các quyết định phù hợp, kịp thời.
- .....

11

## Nội dung

- Tình huống dẫn nhập
- Kho dữ liệu (DW) là gì**
- Các kiến trúc
- Đặc điểm kho dữ liệu
- Câu hỏi thảo luận

12

## Kho dữ liệu (KDL)

- Hệ thống dữ liệu (db system) = CSDL + Hệ QTCSDL
- KDL = Thông tin + phần mềm khám phá thông tin này

HTDL	KDL
<ul style="list-style-type: none"> <li>Xử lý các giao dịch hàng ngày</li> <li>Thêm, xoá, cập nhật, rút trích dữ liệu/thông tin</li> </ul>	<ul style="list-style-type: none"> <li>Hỗ trợ ra quyết định</li> <li>rút trích thông tin để phân tích, để khám phá tri thức, để chọn lựa quyết định</li> </ul>

13

## 2. Định nghĩa 1

- A data warehouse is a system:**
  - retrieves data periodically from the source systems into a dimensional or normalized data store
  - It usually keeps years of **history** and is queried for business intelligence or other analytical activities. It is typically **updated in batches**, not every time a transaction happens in the source system

Vincent Rainardi  
Building a Data Warehouse: With Examples in SQL Server

14

## 2. Định nghĩa khác

"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"

(Ralph Kimball)  
*Building the Data Warehouse, Fourth Edition* (John Wiley, 2005)

"a DW is a **subject-oriented**, **integrated**, **time-variant**, and **nonvolatile** collection of data in support of management's decision making process"

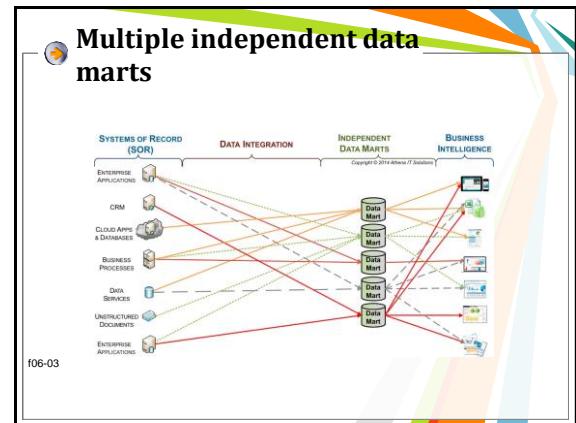
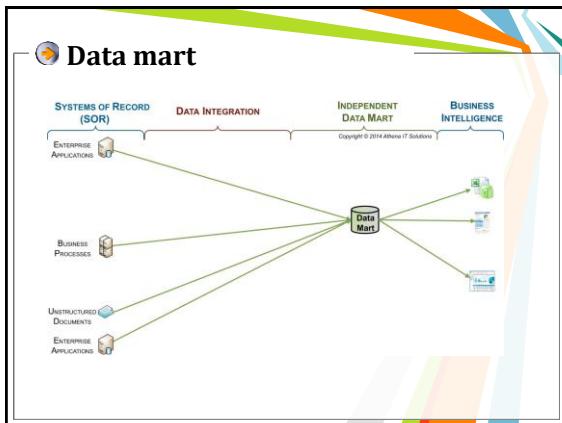
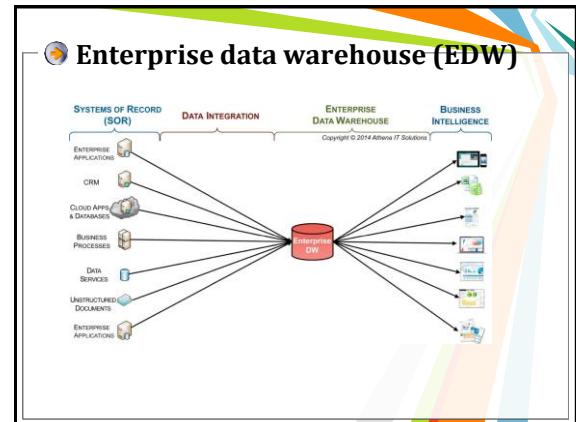
(W.H. Inmon)  
*The Data Warehouse ETL Toolkit* (John Wiley, 2004)

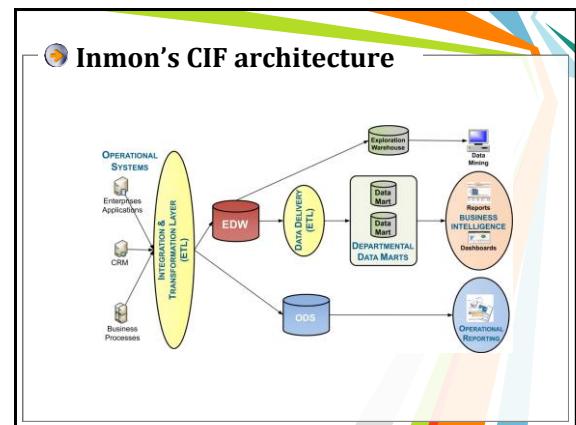
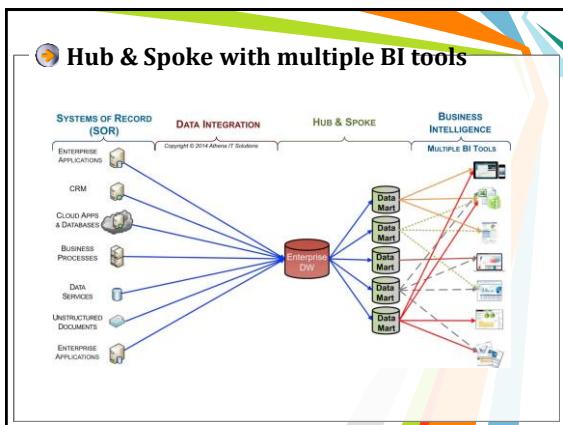
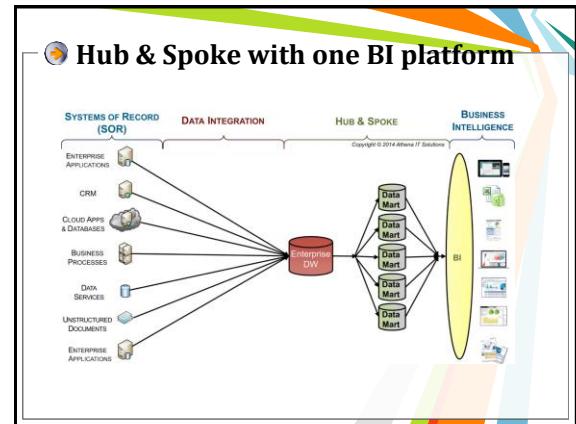
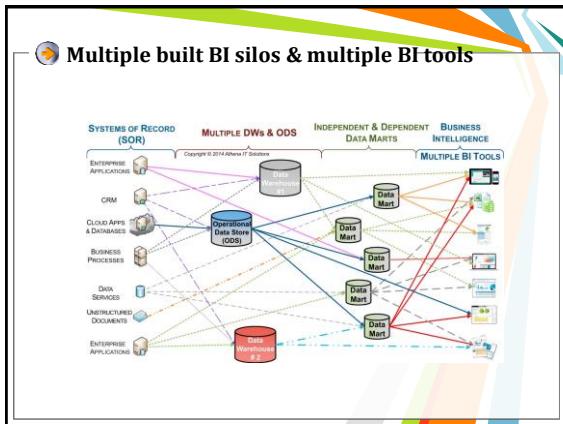
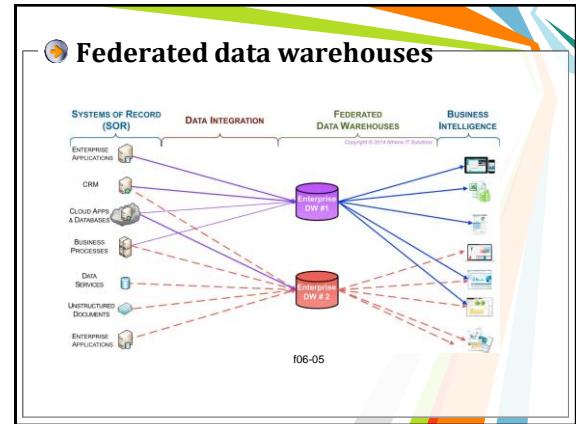
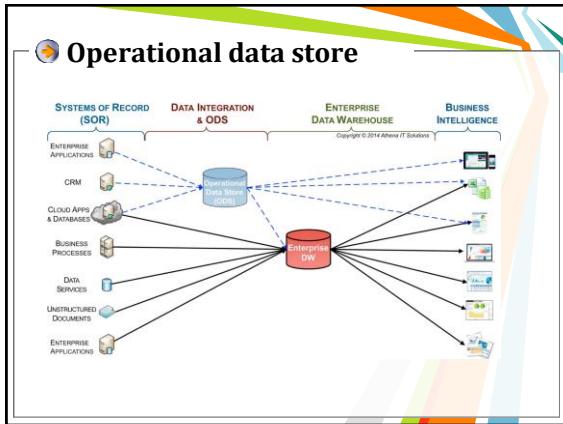
15

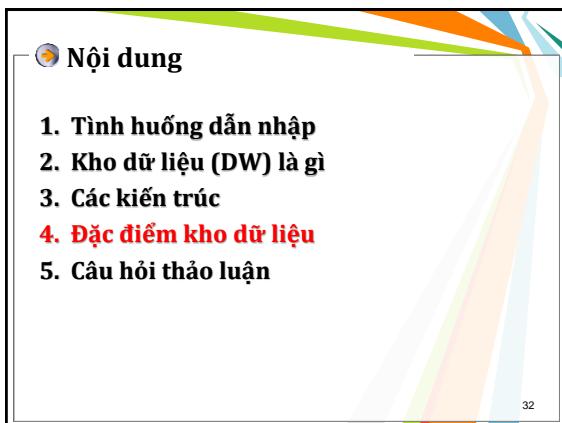
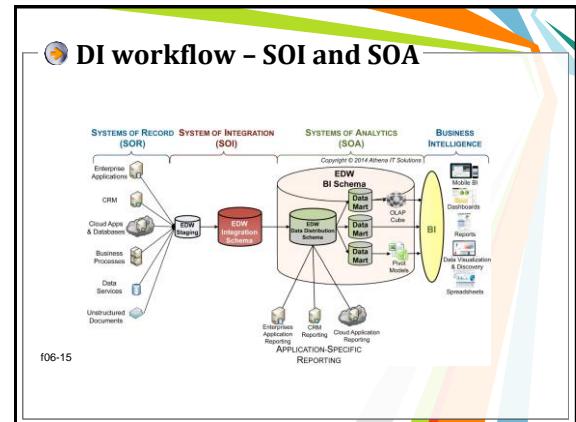
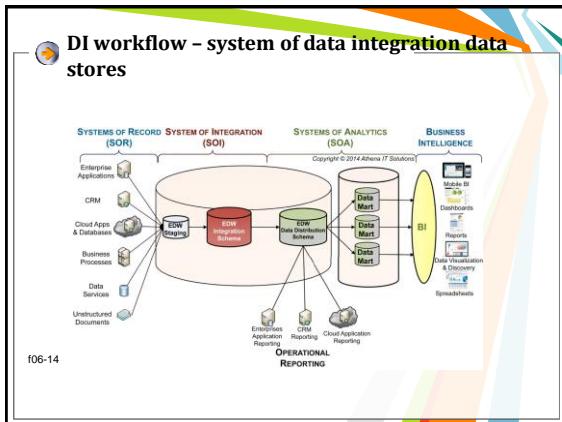
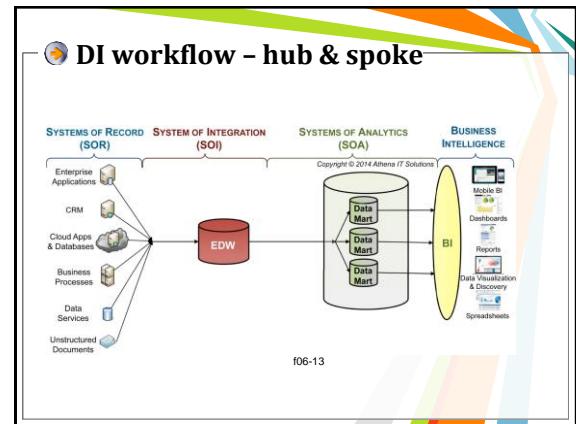
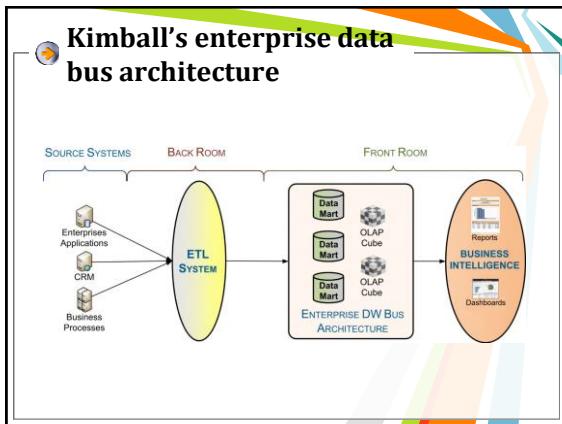
## Nội dung

1. Tình huống dẫn nhập
2. Kho dữ liệu (DW) là gì
3. Các kiến trúc
4. Đặc điểm kho dữ liệu
5. Câu hỏi thảo luận

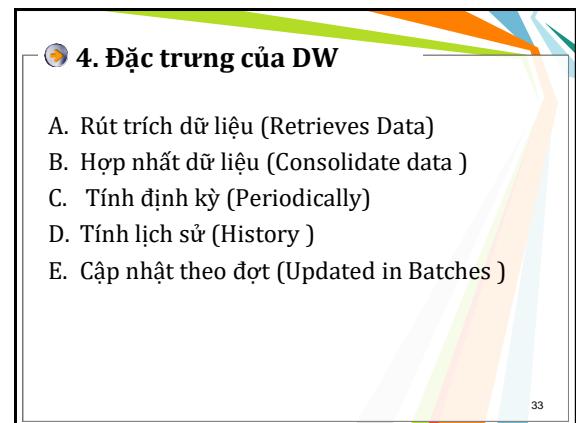
16



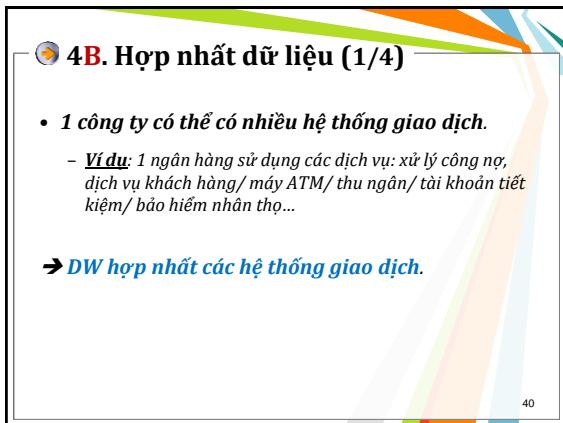
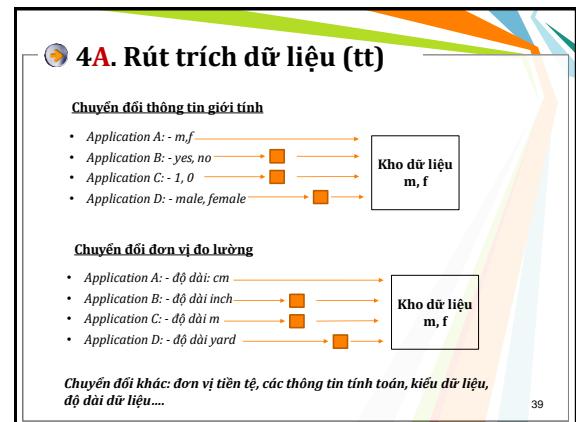
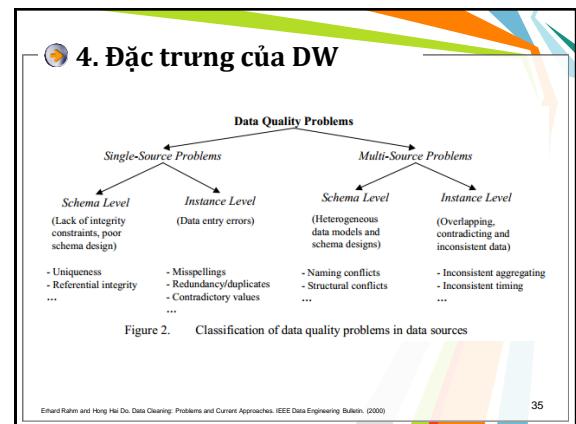
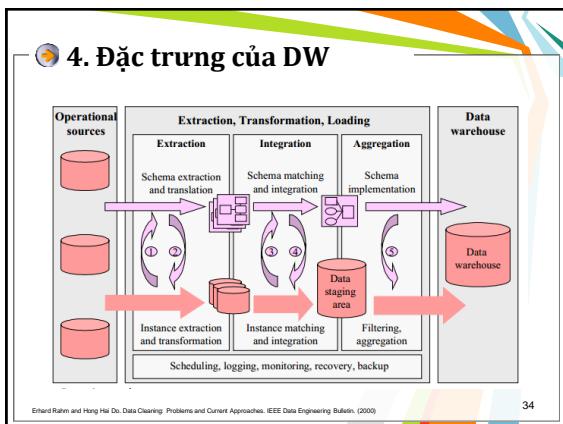




32



33



## 4B. Hợp nhất dữ liệu (3/4)

- c. **Definitions:** Đôi khi tên dữ liệu giống nhau nhưng chứa ý nghĩa khác nhau
  - Ví dụ: tổng giá trị hóa đơn trong system A gồm: thuế, ti lệ giảm, phí vận chuyển; trong khi system B không có phí vận chuyển
  - Ví dụ: cần hợp nhất dữ liệu của khách hàng X trong hệ thống A với dữ liệu khách hàng X trong hệ thống B
- d. **Conversion:** khi hợp nhất dữ liệu đôi khi cần thực hiện một số chuyển đổi bởi vì dữ liệu trong source system có các định dạng khác nhau.
  - Ví dụ: chuyển thời gian giữa các quốc gia, tỉ giá tiền tệ...

42

## 4B. Hợp nhất dữ liệu (4)

- e. **Matching:** quyết định xem một phần dữ liệu trong sysA có giống như trong hệ thống khác không. Đây là công việc quan trọng, vì nếu so khớp sai sẽ làm cho dữ liệu trong DW không chính xác

- Ví dụ: cần quyết định xem liệu khách hàng X ở hai hệ thống có là 1 hay không, nếu so khớp sai, giao dịch của 1 khách hàng sẽ bị trộn lẫn với 1 khách hàng khác

43

## 4.C Tính định kỳ

- ❑ Rút trích và hợp nhất dữ liệu không chỉ thực hiện 1 lần → thực hiện nhiều lần theo khoảng thời gian đều đặn (*chương 3*)
  - Ví dụ: mỗi ngày, mỗi 3 ngày, mỗi tuần...
  - Nếu chỉ rút trích dữ liệu một lần → sau này dữ liệu sẽ quá hạn, không hữu ích
- ❑ Chu kỳ rút trích **lặp lại** được xác định dựa vào các nhu cầu kinh doanh và tần suất thực hiện việc cập nhật dữ liệu từ csdl nguồn
  - Ví dụ: source system: cập nhật 1 lần/ngày → chu kỳ rút trích 1 lần/ngày

44

## 4.D Tính lịch sử

- ❑ **Điểm khác biệt chính của các hệ thống OLTP và kho dữ liệu chính là khả năng và sức chứa để lưu trữ dữ liệu lịch sử (chương 4).**
- ❑ Ví dụ: Ta cần giữ các thông tin trong 2 năm, trường hợp khách hàng muốn truy vấn thông tin, nhưng ta lại không muốn giữ tới 10 năm trên hệ thống hiện tại vì sẽ làm chậm và giảm hiệu suất.
- **Các hệ thống OLTP:**
  - Lưu trữ khoảng 3 năm dữ liệu, phần còn lại lưu vào thiết bị lưu trữ
  - Thông tin danh mục (vd: sản phẩm) thường được cập nhật mô tả thành mới, không lưu lại mô tả cũ
- **Kho dữ liệu**
  - Lưu trữ một lượng dữ liệu rất lớn.
  - Lưu trữ các luồng truy cập website, thông tin mua hàng của 1 hệ thống siêu thị lớn, thông tin của ngành truyền thông (truyền hình, điện báo, điện thoại...)

45

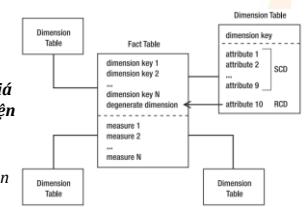
## 4.E Cập nhật theo đợt

- ❑ Dữ liệu từ các hệ thống tác vụ thay đổi liên tục (vài phút, vài giây...) → khó khăn khi phân tích (*chương 3*)
  - ❑ Cập nhật theo thời gian thực (realtime)
    - Cần cài trigger lên csdl lên mỗi bảng của nguồn hoặc
    - Cảnh báo ứng dụng nguồn để ghi dữ liệu vào kho ngay sau khi có thay đổi từ nguồn
    - Nếu nguồn là csdl lớn, cần rút trích dữ liệu từ nhiều bảng → ảnh hưởng đáng kể đến hiệu suất của hệ thống
- Cập nhật theo đợt, sử dụng hệ thống ETL, tại một thời điểm cụ thể**

46

## 4.F Dimensional Data Store (DDS)

- ❑ DDS lưu dữ thông tin của 1 sự kiện kinh doanh, phục vụ cho 1 nhu cầu phân tích (*chương 4*)
  - **Fact table:** chứa các giá trị đo lường của sự kiện kinh doanh
  - **Dimension table:** chứa các ngữ cảnh của sự kiện kinh doanh



47

## 4.F Dimensional Data Store (DDS)

- Ví dụ 1:** sự kiện khách hàng A mua sản phẩm B,C ở một cửa hàng D
  - Fact:** Hóa đơn chứa doanh số bán hàng...
  - Dimension:** khách hàng, cửa hàng, sản phẩm, thời gian mua...

48

## 4.F Dimensional Data Store (DDS)

- Ví dụ 2:**
  - Sự kiện khách hàng A thực hiện giao dịch tại một ngân hàng vào 1 thời điểm...
  - Fact:** bảng GiaoDich chứa (số tiền giao dịch)...
  - Dimension:** khách hàng, tài khoản, thời điểm giao dịch, chi nhánh giao dịch...

49

## Lợi ích

- Một số hệ thống sử dụng DW để tăng lợi nhuận:**
  - Fingerhut:** áp dụng DW và các công cụ data mining để xác định khách hàng tiềm năng và ra quyết định về các chiến dịch quảng cáo
    - <http://www.dmnnews.com/fingerhuts-updated-data-warehouse-increases-profits/article/61617/#>
  - WalMart:** áp dụng DW để tăng lợi ích kinh doanh và hỗ trợ ra quyết định
    - Xem thêm: **Wal-Mart's Data Warehouse**. Patrick (2006)

50

## Thảo luận

51

## Tham khảo

- S.Nagabhushana. Data Warehousing OLAP and data mining. 2006. New Age International (P) Ltd, Publishers
- Vincent Rainardi . *Building a Data Warehouse With Examples in SQL Server*. 2008. Apress.

- Rick Sherman - Business intelligent guide book 2014

62

**Thank You!**

Giảng viên: Hồ Thị Hoàng Vy  
2013, Jan 27

63



### Các loại dữ kiện

- Có 3 loại dữ liệu dữ kiện:
  - Additive
  - Semi-Additive
  - Non-Additive

2

### 1. Additive facts

- Là những dữ kiện có thể được cộng dồn lên qua các chiều liên quan trong KDL
- Ví dụ: Giá bán 1 sản phẩm, số lượng

Date	FactSales	Product
DateKey	DateKey	ProductKey
Year	ProductKey	ProName
Quarter	SalePrices	Category
Month		SubCategory
day		

3

<http://sqlserver-qa.net/2015/06/25/different-types-of-facts-and-fact-tables-in-data-warehouse-design/>

### 2. Non Additive Facts

- Là những dữ kiện không thể được cộng dồn cho bất kỳ chiều nào liên quan tới bảng fact trong KDL
- Ví dụ: tỷ lệ, % lợi nhuận,

Date	FactSales	Customer
DateKey	DateKey	CusKey
Year	ProductKey	CusName
Quarter	TotalSalePrices	
Month		
day	NetProfitMargin	

4

<http://sqlserver-qa.net/2015/06/25/different-types-of-facts-and-fact-tables-in-data-warehouse-design/>

### 3. Semi-Additive Facts

- Là những dữ kiện có thể được cộng dồn dựa trên tất cả các chiều trong KDL ngoại trừ chiều Thời gian
- Ví dụ: Số dư tài khoản

Date	FactAccounts	Customer
DateKey	AccountKey	CusKey
Year	DateKey	CusName
Quarter	TypeKey	
Month	CustomerKey	TypeName
day	Balance	AccountKey

5

<http://sqlserver-qa.net/2015/06/25/different-types-of-facts-and-fact-tables-in-data-warehouse-design/>

### Câu hỏi

- Với mỗi loại dữ liệu dữ kiện, hãy cho ví dụ
- Giải pháp ?

8

## Các loại bảng Facts

- Có thể chia thành 3 loại sau tuỳ theo nhu cầu nghiệp vụ:
  - Transaction Fact Tables
  - Accumulating Snapshot Fact Tables
  - Periodic Snapshot Fact Tables

Ralph Kimball - The Data Warehouse ETL Toolkit

9

## Transaction Fact Tables

- Đây là loại được dùng phổ biến nhất
- Mỗi dòng trong bảng biểu diễn 1 sự kiện cụ thể trong quy trình nghiệp vụ
- Chứa 1 hoặc nhiều khoá ngoại đến các chiều liên quan
- Thường dung đo lường trong môi trường bán lẻ, mức độ chi tiết dữ liệu thường là ngày
- “the facts must be true to the grain”

10

## Accumulating Snapshot Fact Tables

- Loại bảng dữ kiện này được dùng để mô tả các quy trình có sự bắt đầu và kết thúc rõ ràng
  - VD: xử lý đơn đặt hàng, xử lý yêu cầu bồi thường
- Ảnh chụp tích luỹ không phù hợp với các quy trình liên tục, thời gian chạy dài (vd: theo dõi các tài khoản ngân hàng)
- Mỗi bản ghi trong bảng đại diện cho 1 thực thể của quá trình tương ứng và sẽ được cập nhật theo tình trạng của thực thể.

11

## Accumulating Snapshot Fact Tables

- Ví dụ: xét một Sales order: gồm các giai đoạn
  - Order generated
  - Picking order
  - Packing order
  - Shipping order
- Trong bảng fact, ta sẽ có 1 bảng ghi cho 1 order và sẽ được update dựa vào tình trạng của order khi KDL xử lý từng giai đoạn

12

## Accumulating Snapshot Fact Tables

OrderKey	OrderDate	Picking Date_Key	Packed Date_Key	Shipped Date_Key	Days to Pick	Days to pack	Days to shipped	Picker_Key	Packer_Key	Shipper_Key
1234	20150601	Y	Y	Y	0	0	0	0	0	0
		When order placed								
1234	20150601	20150610	0	0	10	0	0	456	0	0
		When Order Picked								
1234	20150601	20150610	20150615	0	10	5	0	456	6578	0
		When Order Packed								
1234	20150601	20150610	20150615	20150617	10	5	2	456	6578	7895
		When order is shipped to customer								

13

## Accumulating Snapshot Fact Tables

Order Date (FK)	Requested Ship Date (FK)	Actual Ship Date (FK)	Delivery Date (FK)	Last Payment Date (FK)	Return Date (FK)	Settlement Date (FK)	Warehouse (PK)	Customer (PK)	Product (PK)	Promotion (PK)	Payment Terms (FK)	Order Number (DD)	Shipment Invoice Number (DD)	Line Number (DD)	Extended List Price (fact)	Promotion Allowance (fact)	Net Invoice Amount (fact)	Amount Paid (fact)	Amount Refunded (fact)	Terms Discount Amount (fact)
-----------------	--------------------------	-----------------------	--------------------	------------------------	------------------	----------------------	----------------	---------------	--------------	----------------	--------------------	-------------------	------------------------------	------------------	----------------------------	----------------------------	---------------------------	--------------------	------------------------	------------------------------

the “standard scenario”

**Figure 6.6** An accumulating snapshot fact table where the grain is the shipment invoice line item.

Ralph Kimball - The Data Warehouse ETL Toolkit

14

## Periodic Snapshot Fact Tables

- Loại này thể hiện bản chụp của quy trình kinh doanh trong 1 khoảng thời gian cụ thể.
- Độ mịn của dữ liệu trong bảng này có thể không ở mức quy trình, mà tổng kết hoạt động trong 1 khoảng thời gian: năm, tháng, quý hoặc tuần

15

## Periodic Snapshot Fact Tables

- VD: bán hàng và giảm giá cho khách hàng trong 1 tháng

Month_Key	Customer_Key	Sales	Discount given
12	1234	40000	400
12	1245	50000	400

This table shows sales and discount given to customer for specific month Period.

16

## Periodic Snapshot Fact Tables

17

Thank You!

TB: Hồ Thị Hoàng Vy  
2015, Jan 27

18

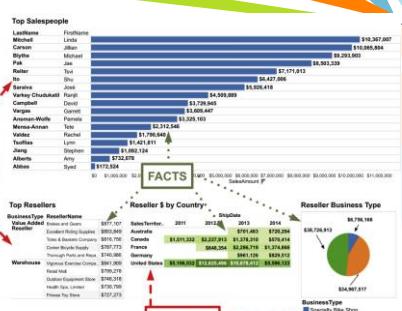
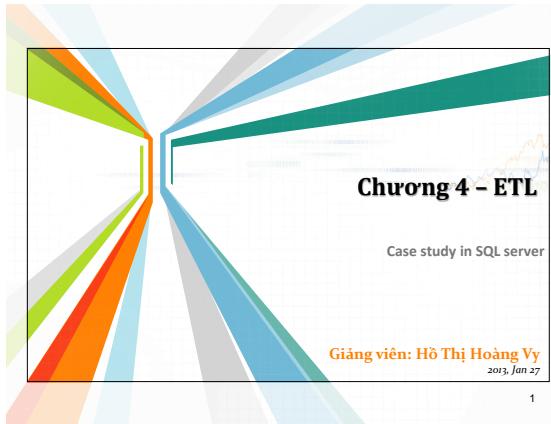


FIGURE 9.16

Managing in a business report

19



## Nội dung

- ❑ Giới thiệu
- ❑ Kiến trúc ETL
- ❑ Chiến lược đổ dữ liệu
  - ✓ Nguồn → stage
  - ✓ Stage → NDS
  - ✓ NDS → DDS
- ❑ Chiều thay đổi chậm - cài đặt
- ❑ Thảo luận

2

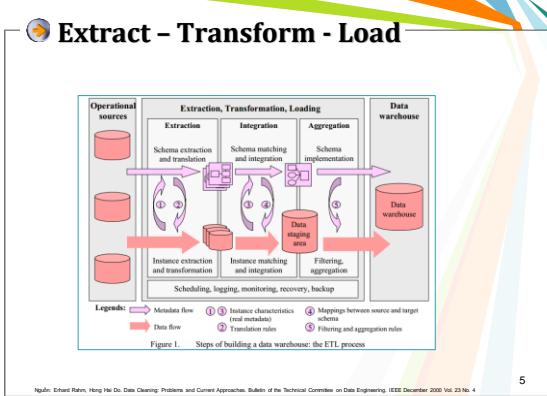
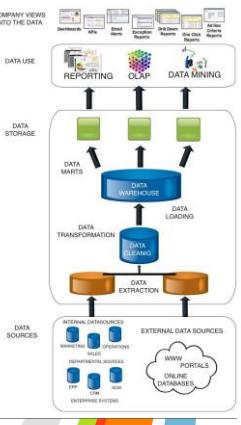
## Nhắc lại

- **Quá trình ETL làm gì?**
  - Quá trình rút trích và chuyển đổi dữ liệu từ hệ thống nguồn và đổ vào kho dữ liệu
- **Hai vấn đề cơ bản của 1 quá trình ETL:**
  - Không thất thoát thông tin (leakage)
  - Khả năng phục hồi (recoverability)
    - Tính chịu lỗi cao

3

## Giới thiệu

- Xem định nghĩa (Chương 1)
  - **Extract:** rút trích dữ liệu liê i
  - **Transform:** biến đổi các dữ target system; suy diễn giá liệu hợp lệ, làm sạch dữ liệu
  - **Load:** tải dữ liệu vào target kẽ.



5

Nguồn: Erhard Raab, Hong Hai Do, Data Cleaning: Problems and Current Approaches, Bulletin of the Technical Committee on Data Engineering, IEEE December 2010 Vol 23 No 4

## Extract - Transform - Load

- ETL - Phát hiện và hiệu chỉnh các vấn đề về chất lượng dữ liệu, nạp các lượng dữ liệu lớn vào kho dữ liệu

- A. Chất lượng dữ liệu (data quality)**
- B. Nạp và cập nhật (Load & refresh)**

The process of identifying and correcting dirty data.  
Dirty data means incomplete, wrong, duplicate, or out-of-date data

6

## Data Cleansing & matching

### A. Chất lượng dữ liệu (data quality)

- 1 nguồn
  - Thiếu các RBTW
  - Thiết kế lược đồ quá tệ
  - Tính duy nhất
  - Ràng buộc tham chiếu
  - Dữ liệu nhân bản hoặc dữ thừa
  - Sai chính tả
  - ....

- Nhiều nguồn
  - Không đồng nhất mô hình dữ liệu và thiết kế lược đồ
  - Xung đột cách đặt tên
  - Xung đột về cấu trúc
  - Dữ liệu chồng chéo, không nhất quán
  - Các thống kê không nhất quán
  - Thời gian không nhất quán
  - ...

7

## Extract - Transform - Load

### Chất lượng dữ liệu (data quality)

#### Ví dụ:

- State = "California" & country = "Canada"
- Không nhất quán khi biểu diễn cùng dữ liệu ('CA', 'California')
- Số điện thoại không hợp lệ
- Thiếu thông tin trong dữ liệu: Email sai định dạng (thiếu @)
- Mã sản phẩm không tồn tại
- Giá: 0\$ / quyển sách
- ....

8

## Extract - Transform - Load

### Chuỗi:

- Tìm thấy tên city "Los Angles" nhưng không tồn tại trong bảng city
- Nhận diện Robert Peterson & Bob Peterson là 1 người

### Số:

- 5 vs 6
- 5.029 vs 5.03
- Có thể dùng phép "=" để so khớp dữ liệu số. Ví dụ: If A=B...

### Ngày tháng:

- 03/01/2008 vs 01/03/2008 ?

### Logic matching:

- Chính xác (exact), gần đúng (fuzzy), dựa trên luật (rule based)

9

## Extract - Transform - Load

### Data profiling tools

- Ví dụ: (tên, địa chỉ) phải duy nhất
  - Xác định tính duy nhất trong csdl
  - Xác định mức độ vi phạm
    - Một trong hai Tên/Địa Chỉ thiếu thông tin
- Khám phá các luật hay các thuộc tính trong csdl cho trước.
  - Xác định cột/ tập các cột nào là khóa (duy nhất) trong source
  - Xác định các phụ thuộc hàm (cùng zipcode → !cùng state)
  - Xác định vi phạm RBKN
  - ....

10

## Extract - Transform - Load



- VD: SSIS tool

11

## Extract - Transform - Load

### Cải thiện chất lượng dữ liệu:

- Rút trích cấu trúc (extracting structure)
  - Cho 1 dòng dữ liệu văn bản → phân tích cú pháp nạp vào các thuộc tính tương ứng trong KDL
- Bỏ trùng lặp (deduplication)
  - Tính độ tương đồng giữa hai đối tượng, ví dụ: "Robert" và "Robet"
  - Từ đồng nghĩa
    - Fuzzy matching
- Hầu hết các nhà sản xuất công cụ ETL đều hỗ trợ "fuzzy matching" và "deduplication"

12

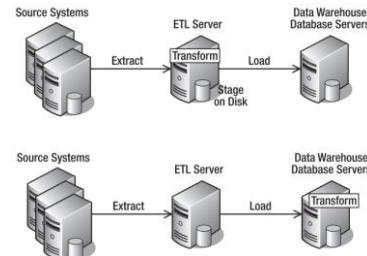
## Extract - Transform - Load

### B. Nạp và cập nhật (Load & refresh)

- Đưa dữ liệu từ nguồn vào kho sao cho: nhanh & ít ảnh hưởng hiệu suất nhất có thể.
  - Xác định dữ liệu cần nạp
- Kỹ thuật:
  - Trigger
  - Sniff transaction logs
  - Bulk-loading
  - Load into partition
  - checkpoint

14

## Kiến trúc ETL



15

## Kiến trúc ETL

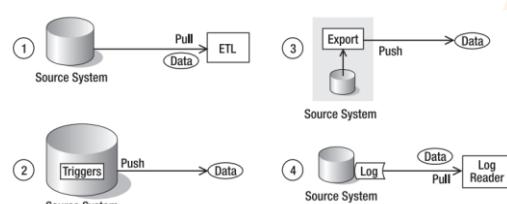
### A. Phân loại theo đối tượng chủ động thực hiện rút trích

#### ➔ 4 phương pháp ETL:

1. Lấy dữ liệu bằng cách truy vấn đều đặn dữ liệu nguồn
2. Cài trigger trong cơ sở dữ liệu nguồn để lấy dữ liệu mỗi khi nguồn có sự thay đổi
3. Lập lịch lấy dữ liệu từ nguồn
4. 1 chương trình đọc tập tin nhật ký để xác định các thay đổi trong dữ liệu

16

## Kiến trúc ETL



17

## Kiến trúc ETL

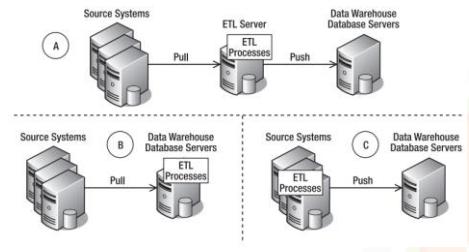
### A. Phân loại theo nơi thực hiện quá trình ETL

#### ➔ 3 phương pháp ETL:

1. Thực hiện quá trình ETL tại 1 server riêng (trung gian giữa nguồn và kho dữ liệu)
2. Thực hiện quá trình ETL trong kho dữ liệu
3. Quá trình ETL thực hiện tại host của hệ thống nguồn

18

## Kiến trúc ETL



19

## Stage loading

- Source**
  - Cơ sở dữ liệu quan hệ
  - Flat files
  - Các loại nguồn khác (xml, webservice...)
- Khi rút trích dữ liệu từ nguồn → không làm gián đoạn nguồn quá nhiều
- Sau khi rút trích dữ liệu → đưa ngay vào kho dl, không nên lưu trữ tạm thời vào 1 csdl hay tập tin khác.

20

## Rút trích từ Flat files

- Tập tin:**
    - Ví dụ

```
001|Nguyễn Ngọc Thảo|Hành chính
002|Trần Thanh Toản | Tài vụ
003|Lê Thành Nhì | Kỹ Thuật
```
  - Hiệu xuất nhập/xuất dữ liệu nhanh
  - Bulk insert trong SQL command cho phép load dữ liệu từ flat file nhanh chóng
- bulkinsert table1 from 'file1' with (fieldterminator = '|')

21

## Rút trích từ Flat files

- Các lưu ý:**
  - Phải thỏa thuận cấu trúc tập tin với DBA của hệ thống nguồn
    - Đặt tên,
    - Kí tự phân cách...
  - Có quyền truy cập tập tin nguồn để:
    - xoá,
    - lưu trữ,
    - điều khiển lỗi,
    - tân xuất load dữ liệu
    - ....

22

## Rút trích từ CSDL quan hệ

### Gồm 3 chiến lược

- 1. Incremental extract**
- 2. Fixed range**
- 3. Whole table every time**

23

## 1. Incremental extract

- Đặc điểm:** Chỉ rút trích những thay đổi so với lần rút trích gần đây nhất
- Dòng mới thêm
  - Dòng mới xoá
  - Dòng được cập nhật
- Dựa vào các thuộc tính:** nhãn thời gian, thuộc tính tự tăng, ngày giao dịch, triggers, kết hợp các thuộc tính này.

24

## 1. Incremental extract

**Giải pháp 1:** sử dụng nhãn thời gian **Created & last updated**, Order status

- Created, last updated:** ngày tạo, ngày cập nhật sau cùng
  - Order status:** trạng thái của dòng ghi nhận dòng đã xoá (không xoá thực sự)
  - Mỗi khi 1 dòng có thay đổi (update, delete) → cập nhật timestamp
- ➔ Dò tìm Thêm, xoá, cập nhật = dò tìm thêm, cập nhật

Order ID	Order Date	Some Columns	Order Status	Created	Last Updated
45435	10/16/2007	Some Data	Canceled	10/16/2007 11:23:35	10/17/2007 16:19:03
...					

25

## 1. Incremental extract

### Trường hợp thêm & cập nhật

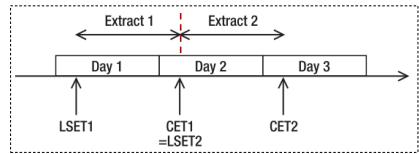
- CET: Thời điểm rút trích hiện tại
- LSET: Thời điểm rút trích thành công gần nhất

#### Các bước:

- Lấy thông tin LSET được lưu trong metadata
- Lấy CET: thời gian khởi động ETL package
- Rút trích dữ liệu:  
select \* from order\_header where (created  $\geq$  LSET and created  $<$  CET) or (last\_updated  $\geq$  LSET and last\_update  $<$  CET).
- Cập nhật LSET = CET

26

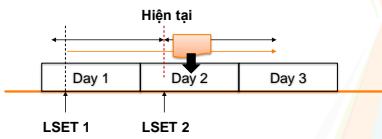
## Thảo luận



- Tại sao phải lấy CET ở bước 2, không phải là B4?
- Tại sao cần chặn trên CET?

27

## Thảo luận



#### Câu 1

- Hiện tại cần thực hiện ETL.
  - Lấy dữ liệu từ lần thực hiện ETL trước đó (thời điểm LSET1) tới hiện tại
- Giả sử tại lúc đang ETL, có 1 hóa đơn được tạo ra**
  - Hoá đơn vẫn được xử lý bởi ETL
  - ETL lần 2, lấy tất cả hóa đơn từ sau lần ETL gần nhất  $\rightarrow$  rút trích lại hóa đơn tạo ra sau từ lần ETL1  $\rightarrow$  2 lần

28

## Thảo luận

**Giải pháp 2:** sử dụng Order date (thay cho created), last updated, order status, ràng buộc nghiệp vụ.

**Ví dụ:** với ràng buộc hoá đơn phải được thêm vào hệ thống trong vòng 28 ngày kể từ ngày giao dịch

#### Các bước:

- Lấy thông tin LSET được lưu trong metadata
- Lấy CET: thời gian hiện tại của hệ thống
- Rút trích dữ liệu:  
select \* from order\_header where order\_date  $\geq$  (LSET - 28 days) and orderdate  $<$  CET or (last\_updated  $\geq$  LSET and last\_update  $<$  CET).
- Cập nhật LSET = CET

29

## 1. Incremental extract

**Giải pháp 3:** sử dụng thuộc tính tự tăng (identity), last updated, order status

- Trường hợp dò tìm thêm mới**
  - Lấy ID được rút trích sau cùng nhất (LSEI) từ csdl metadata
  - Lấy max(orderID) từ bảng hoá đơn, gán vào CEI (ID rút trích hiện tại)
  - Lấy tập các dòng nằm giữa LSEI và CEI như sau:
  - Select \* from order\_header where order\_id  $\geq$  LSEI and order\_id  $<$  CEI
  - Gán LSEI mới = CEI
- Trường hợp dò tìm cập nhật:** tương tự giải pháp 1 (sử dụng last updated)

30

## 1. Incremental extract

**Giải pháp 4:** không có order status  $\rightarrow$  xoá thực sự khỏi bảng

$\triangleright$  không thể dò tìm delete dựa vào nhãn thời gian

#### ☐ Có 2 giải pháp tìm dòng bị xoá

- So sánh khoá chính giữa bảng nguồn (source) và bảng trong DW.
- Sử dụng trigger

31

## 1. Incremental extract

Có thể dùng trigger để dò tìm các dòng thêm mới, cập nhật, xoá mà không dùng các nhãn thời gian?

- **Nhận xét**

- Trigger là cách tốt nhất để phát hiện thay đổi trong dữ liệu nguồn (insert, update, delete).
- **Khuyết:** cần thận khi cài trigger để ghi nhận dòng bị thay đổi vào bảng event.
  - Chỉ ghi nhận khoá chính, không ghi nhận được dữ liệu thay đổi nhiều lần → lưu cả dòng vào bảng
  - Mất thời gian để thực hiện xong 1 giao tác → phụ thuộc vào độ phức tạp của trigger & hệ thống OLTP

32

## 2. Fixed range

(Rút trích vùng cố định)

- **Tình huống:**

- Không thể rút trích toàn bộ bảng vì khối lượng quá lớn
- Không thể rút trích incremental do
  - Không có thuộc tính nhãn thời gian, nhãn thời gian không tin cậy
  - Thuộc tính tự tăng không tin cậy
  - Không thể cài trigger trên bảng source

33

## 2. Fixed range

(Rút trích vùng cố định)

- **Giải pháp:** rút trích 1 số lượng chính xác các dòng hoặc theo 1 khoảng thời gian cụ thể dựa vào ràng buộc nghiệp vụ.

- **Ví dụ:** việc xử lý dữ liệu sẽ được hoàn tất vào cuối mỗi tháng dựa vào ngày giao dịch → rút trích dữ liệu vào cuối mỗi tháng & chỉ lấy về dữ liệu trong tháng đó.

34

## Whole table

- **Tình huống:**

- Kích thước bảng nhỏ
- Không có nhãn thời gian
- Không có thuộc tính tự tăng
- Không có ràng buộc nghiệp vụ

- **Giải pháp**

- Rút trích toàn bộ dữ liệu nguồn

35

## Rút trích từ các nguồn khác

- XML
- spreadsheet files (Excel)
- Web logs
- Nhật ký giao tác CSDL
- Binary file
- Webservice
- Emails
- .....

36

## Stage → NDS

- **Lưu ý các vấn đề:**

- Chuẩn hoá**
  - NDS chuẩn càng cao → phục vụ nhiều nhu cầu phân tích
- Chuyển đổi dữ liệu**
  - Cần thực hiện một số biến đổi để đồng nhất các nguồn với định dạng của NDS
- Quản lý khoá**
  - Lưu vết nguồn dữ liệu
  - Xác định sự tồn tại của dữ liệu

37

## Chuẩn hoá

- Nguồn

store_number	store_name	store_type
1805	Perth	Online
3409	Frankfurt	Full Outlet
1014	Strasbourg	Mini Outlet
2236	Leeds	Full Outlet
1808	Los Angeles	Full Outlet
2903	Delhi	Online

NDS

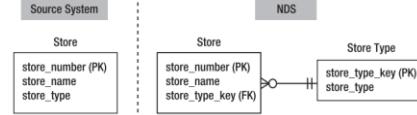
store_number	store_name	store_type_key
1805	Perth	1
3409	Frankfurt	2
1014	Strasbourg	3
2236	Leeds	2
1808	Los Angeles	2
2903	Delhi	1



38

## Chuẩn hoá

- Nâng chuẩn



- NDS được chuẩn hoá từ 3NF trở lên
  - Source system có thể không được chuẩn hoá
- cần chuẩn hoá dữ liệu để đáp ứng cấu trúc trong NDS

39

## Chuẩn hoá

- Quản lý khoá

- Khoá tự nhiên - natural key (NK)

- Khoá chính của 1 dòng trong source system
- Khi đổ dữ liệu từ stage vào NDS → cần 1 khoá đại diện

- Khoá đại diện - surrogate key (SK)

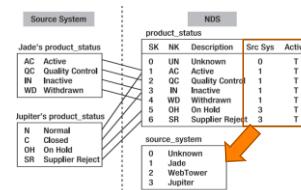
- Định danh 1 dòng trong kho dữ liệu
- Là khoá chính của bảng chiều trong DDS
- Là số nguyên, bắt đầu từ 0, dùng để liên kết các fact và dimension

40

## Chuẩn hoá

- Quản lý khoá

- Ví dụ



41

## NDS → DDS

- Incremental loading

- Các bảng trong DDS không chuẩn hoá
- Giống như NDS, khi load dữ liệu vào DDS cần thực hiện thao tác UPSERT
  - Insert & update dữ liệu nguồn phụ thuộc vào sự tồn tại của dữ liệu trong kho.
  - NDS đã lưu các nhãn thời gian → biết được thời gian cập nhật sau cùng → chỉ load dữ liệu có thay đổi trong NDS từ lần ETL gần nhất.

42

## NDS → DDS

- Key management

- Kiến trúc NDS + DDS:

- khoá đại diện được lưu và quản lý trong NDS, giúp cho việc UPSERT đơn giản hơn
- Đảm bảo tất cả DDS có sự đồng nhất về khoá đại diện

- Kiến trúc ODS + DDS:

- Chỉ có 1 DDS → không có vấn đề đồng bộ khoá đại diện giữa các DDS
- Quản lý khoá trong DDS tương tự kiến trúc NDS + DDS

43

## NDS → DDS

- Denormalization:**

- Các bảng chiều trong DDS là phi chuẩn, trong khi NDS lại chuẩn hóa → cần join các bảng trước khi nạp vào DDS (ví dụ: product, product\_type)

44

## Chiều thay đổi chậm

- Tình huống**

ID	MakH	HoTen	NgheNghiep
1	KH01	Tài	TaiXe
2	KH02	Vân	ThuKi

3 năm  
Khách hàng có makh = 1,  
Thực hiện 1000 giao dịch.

ID	MakH	HoTen	NgheNghiep
1	KH01	Tài	GiamDoc
2	KH02	Vân	ThuKi

➔ Thống kê doanh thu bán hàng theo nghề nghiệp còn đúng hay không?

45

## Chiều thay đổi chậm

- Ý nghĩa:**

- Giá trị của các chiều tồn tại trong 1 thời gian dài → khi thay đổi ảnh hưởng đến kết quả phân tích
- Cần lưu trữ giá trị lịch sử của các chiều

- Phương pháp**

- Ghi chồng giá trị cũ và mới
- Lưu trữ giá trị cũ
  - Theo dòng
  - Theo cột

46

## Chiều thay đổi chậm

- Ví dụ:** store 7 ở region 1 chuyển sang region 2

- Cần lưu trữ thông tin lịch sử của cửa hàng: đã từng ở region 1

**2A. Thêm 1 dòng mới và cập nhật tính tình trạng**

Table 5-6. Storing Historical Information As a Row

key	store	region	status
1	7	1	expired
2	7	2	active

47

## Chiều thay đổi chậm

- Nghề nghiệp: thay đổi chậm

ID	MakH	HoTen	NgheNghiep	TinhTrang
1	KH01	Tài	TaiXe	Active
2	KH02	Vân	ThuKi	Active

3 năm  
Khách hàng có makh = 1,  
Thực hiện 1000 giao dịch.

ID	MAKH	HoTen	NgheNghiep	TinhTrang
1	KH01	Tài	TaiXe	Expired
2	KH02	Vân	ThuKi	Active
3	KH01	Tài	GiamDoc	Active

48

## Chiều thay đổi chậm

- 2B:** Lưu trữ vào 1 cột mới, và thêm cột effective\_date

Table 5-7. Storing Historical Information As a Column

key	store	current_region	old_region	effective_date
1	7	2	1	11/18/2007

49

- Câu hỏi:**

- Nếu cửa hàng lại chuyển vị trí thì lưu trữ thế nào?

## Chiều thay đổi chậm

- Nghề nghiệp: thay đổi chậm

ID	MAKH	HoTen	NgheNghiep	NgheMoi	NgayHL
1	KH01	Tài	TaiXe		1/1/12
2	KH02	Vân	ThuKi		1/1/12

3 năm  
Khách hàng có makh = 1,  
Thực hiện 1000 giao dịch.



ID	MAKH	HoTen	NgheNghiep	NgheMoi	NgayHL
1	KH01	Tài	TaiXe	GiamDoc	1/1/15
2	KH02	Vân	ThuKi		1/1/12

50

## Chiều thay đổi chậm

### Nhận xét

- Giải pháp 1:** ghi chồng giá trị cũ, mới → thông tin cũ sẽ bị mất
- Giải pháp 2a:** linh động hơn, lưu trữ được giá trị lịch sử của chiều, có thể lưu nhiều phiên bản, không cần thay đổi cấu trúc bảng
- Giải pháp 2b:** không linh động, chỉ sử dụng khi biết trước số lần thay đổi cố định, giá trị rất hiếm khi thay đổi

51

## Chiều thay đổi nhanh (thoái hóa)

### Lưu trữ thế nào?

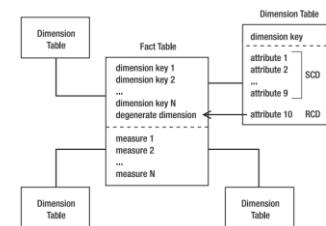
#### 1. Fact table

- Ví dụ: bảng chieu có 10 chieu, 1-> 9 thay đổi chậm, 10 thay đổi nhanh
  - Bảng chieu 10 thuộc tí, chuyển sang bảng fact tat ca cac thuộc tính của chieu nay(thuộc tính này không là khóa của chieu)

#### 2. Nếu có dùng thông tin đó để phân tích → tách ra thành 1 chiều riêng???

52

## Chiều thay đổi nhanh



53

## Thảo luận

### Code để nạp dữ liệu vs dung công cụ?

55

## Quy trình thiết kế

- Top-down, bottom-up approaches or a combination of both
  - Top-down:** Starts with overall design and planning (mature)
  - Bottom-up:** Starts with experiments and prototypes (rapid)
- From software engineering point of view
  - Waterfall:** structured and systematic analysis at each step before proceeding to the next
  - Spiral:** rapid generation of increasingly functional systems, short turn around time, quick turn around
- Typical data warehouse design process
  - Choose a **business process** to model, e.g., orders, invoices, etc.
  - Choose the **grain (atomic level of data)** of the business process
  - Choose the **dimensions** that will apply to each fact table record
  - Choose the **measure** that will populate each fact table record

58

## Tham khảo

1. Vincent Rainardi . *Building a Data Warehouse With Examples in SQL Server. 2008.* Apress.
2. Claudia Imhoff, Nicholas Galembo, Jonathan G. Geiger. *Mastering data warehouse design – Relational and dimensional techniques. 2003.* Wiley Publishing, Inc

59

**Thank You!**

Giảng viên: Hồ Thị Hoàng Vy  
2015, Jan 27

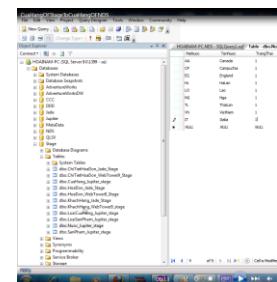
60

## Thảo luận

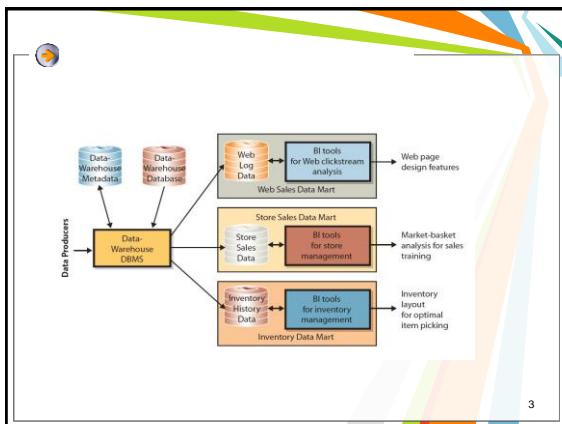
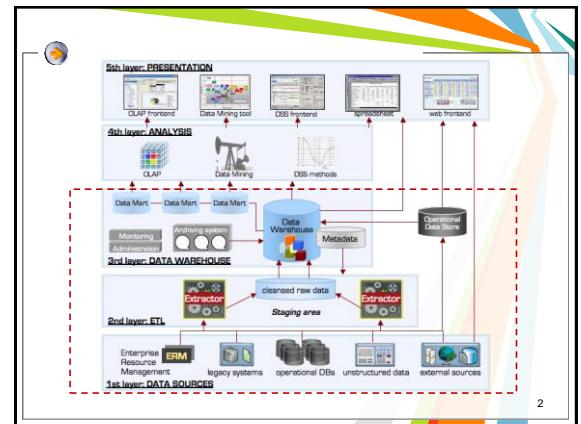
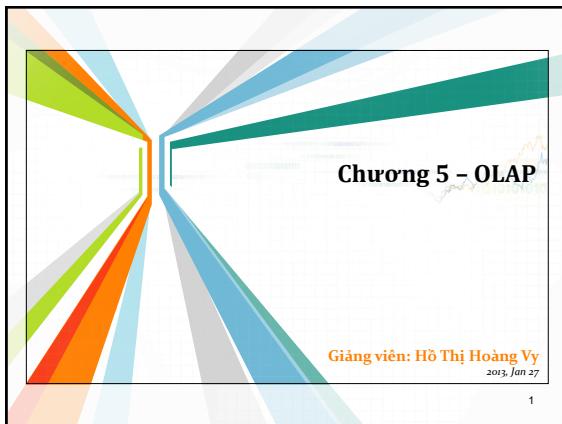
- ETL case study (chương 2)
  - Film demo: stagedb đã được tạo sẵn các table của từng nguồn, đặt tên: stage\_Jade\_tableName
  - Vậy có thể ko cần tạo sẵn các table trong stage mà tự phát sinh lúc ETL?
- Đồ án thực hành
  - Dữ liệu csv, chuỗi cách nhau dấu "," → nên xử lý file csv trước hay sử dụng tool

61

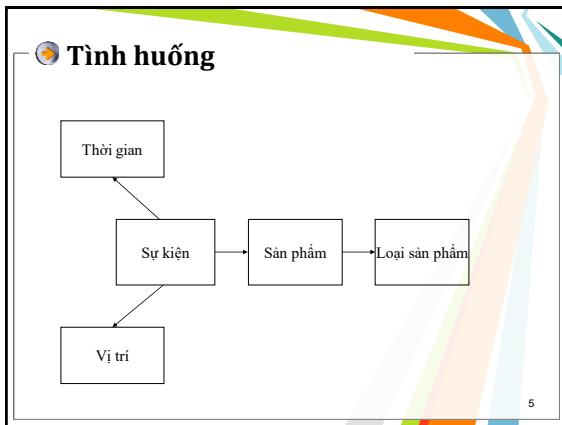
## Thảo luận



62



- ## Nội dung
- Khái niệm cơ bản
    - Dữ liệu đa chiều – CUBE
    - DATA MART
  - Truy vấn dữ liệu với OLAP (ROLAP, MOLAP, HOLAP)
  - Ngôn ngữ MDX
- 4



## Tình huống

	T1	T2	T3	T4
Bánh mì			6	17
Phô mai	6	16	6	8
Cá hộp	8	25	21	

Số lượng sản phẩm đã bán theo mỗi tháng

Thời gian →  
Sản phẩm ↑

6

## Tình huống

- Thống kê số lượng **hàng hoá** của mỗi **thành phố** theo từng **tháng**

		T1	T2	T3	T4
HCM	Bánh mì			3	10
	Phô mai	3	16	6	
	Cá hộp	4	16	6	
Hà Nội	Bánh mì			3	7
	Phô mai	3			8
	Cá hộp	4	9	15	

7

## Tình huống

Thống kê lợi nhuận và số lượng sản phẩm bán ra tại mỗi thành phố theo từng tháng

		Tháng 1		Tháng 2		Tháng 3		Tháng 4	
		Lợi nhuận	SL bán	Lợi nhuận	SL bán	Lợi nhuận	SL bán	Lợi nhuận	SL bán
HCM	Bánh mì					<b>7.44</b>	<b>3</b>	<b>24.80</b>	<b>10</b>
	Phô mai	<b>7.95</b>	<b>3</b>	<b>42.40</b>	<b>16</b>	<b>15.90</b>	<b>6</b>		
	Cá hộp	<b>7.32</b>	<b>4</b>	<b>29.98</b>	<b>16</b>	<b>10.98</b>	<b>6</b>		
Hà Nội	Bánh mì					<b>7.44</b>	<b>3</b>	<b>17.36</b>	<b>7</b>
	Phô mai	<b>7.95</b>	<b>3</b>					<b>21.20</b>	<b>8</b>
	Cá hộp	<b>7.32</b>	<b>4</b>	<b>16.47</b>	<b>9</b>	<b>27.45</b>	<b>15</b>		

8

## 1a. CSDL đa chiều - CUBE

**Lược đồ đa chiều (multidimensional schema)**

- Giống như 1 Rubik's Cube.
- Mỗi ô (cell) của Cube là 1 giá trị đo (measure) thực sự.
- Lược đồ sao và bông tuyết thông thường được lưu trong các csdl quan hệ, còn lược đồ đa chiều được lưu trữ trong csdl đa chiều được gọi là (online analytical processing) OLAP cube.
- CSDL đa chiều lưu trữ và tổng hợp (aggregate) dữ liệu tại các mức khác nhau trong phân cấp (hierarchy), cho phép "roll up – drill down" dữ liệu (di xuống cấp thấp nhất hoặc chi tiết nhất của dữ liệu hoặc xem dữ liệu ở cấp tổng hợp cao nhất..). Các csdl này được thiết kế đặc biệt dựa trên các mảng đa chiều

9

## 1a. CSDL đa chiều - CUBE

- Là 1 hình thức của cơ sở dữ liệu trong đó
  - Dữ liệu được lưu trữ trong các ô
  - Vị trí của mỗi ô được xác định bởi các chiều
  - Mỗi ô đại diện cho 1 **sự kiện**
  - Giá trị của chiều cho biết khi nào và ở đâu xảy ra sự kiện này

10

## 1a. CSDL đa chiều - CUBE

- CSDL đa chiều có  $\geq 4$  chiều trở lên được gọi là **HyperCube**

11

## 1a. Cơ sở dữ liệu đa chiều

- Sử dụng điển hình trong BI, đặc biệt là :
  - Xử lý phân tích trực tuyến (OLAP)
  - Khai thác dữ liệu (data mining).
- Nhiệm vụ:**
  - Khai thác số liệu tổng hợp: phân tích các hoạt động kinh doanh, lập kế hoạch, ra quyết định
  - Thể hiện kết quả trực quan: thuận lợi cho người không chuyên CNTT

12

## 1a. Cơ sở dữ liệu đa chiều

- Ưu điểm**
  - ít tốn không gian và hiệu quả tốt hơn DDS
  - Tối ưu cho các phép toán slice, dice
- Khuyết điểm** (so với sử dụng csdl quan hệ)
  - Cần thời gian xử lý để load dữ liệu
  - Tính toán lại giá trị thống kê
  - Khi nguồn liên quan được cập nhật → các ô thống kê cần được tính toán lại (không theo thời gian thực)
  - Tính co giãn: không phù hợp khi số chiều lớn, dữ liệu lớn (nhiều terabytes)

13

## 1a. Cơ sở dữ liệu đa chiều

2 chiều

	C1	C2
Bánh mì	100	100
Phô mai	150	350
Sữa chua	200	300

Biểu diễn dữ liệu 2 chiều:  
Sự kiện A: khách hàng C1 mua 1 sản phẩm P1

14

## 1a. Cơ sở dữ liệu đa chiều

Doanh số bán hàng theo sản phẩm, địa bàn, thời gian

	Quận 1			Quận 2			Quận 3			
	SP A	SP B	SP C	SP A	SP B	SP C	SP A	SP B	SP C	
2010	100	200	150	50	100	80	....	70	30	50
2009										

15

## 1b. Data mart

- Kho dữ liệu hướng chủ đề** (ví dụ: sales, finance, marketing...)
  - Quy mô nhỏ
  - Lưu trữ dữ liệu chuyên về 1 lĩnh vực, chuyên ngành
- Có thể được hình thành từ việc rút trích 1 **tập con của kho dữ liệu hoặc xây dựng độc lập** và kết nối các data mart tạo thành kho dữ liệu.
- Một số tác giả sử dụng các từ như: **data warehouse, cube, OLAP system, DSS** (hỗ trợ ra quyết định) thay thế cho nhau.

16

## 2. Khai thác dữ liệu đa chiều

- RDBMS: quản lý csdl quan hệ
- MDBMS: quản lý csdl đa chiều
  - OLAP servers
  - Cube engine
- Một số MDBMS:**
  - Microsoft SQL Server Analysis Services (SSAS)**
  - Hyperion Essbase (Oracle)
  - Cognos PowerCube
  - ....

17

## OLAP (online analytical processing)

- OLAP: hoạt động dùng để phân tích CSDL
- OLAP tools: công cụ hỗ trợ phân tích số liệu trực tuyến
  - khai thác số liệu tổng hợp: phân tích hoạt động kinh doanh, lập kế hoạch, ra quyết định, ...
  - Thể hiện kết quả trực quan: thuận lợi cho người không chuyên CNTT
- OLAP cube ~ Multidimensional database

20

## OLAP

- Phân tích dữ liệu giao dịch kinh doanh trong các hệ thống tác vụ như ERP hoặc trong các ODS có được xem là OLAP?
- Hoạt động tương tác 1 chiều: đọc, tạo báo cáo có được xem là OLAP?

➔ Không là OLAP

21

## OLAP

- OLAP phải có sự tương tác
- Các công cụ hỗ trợ BI: report, OLAP, data mining
- OLAP sử dụng
  - MOLAP - multidimensional online analytical processing
  - ROLAP - relational online analytical processing
  - HOLAP - Hybrid Online Analytical Processing

22

## 2. Khai thác dữ liệu đa chiều

### Các mô hình OLAP

- ROLAP** - relational OLAP: lưu trữ các giá trị thống kê được tạo và lưu giữ trong csdl quan hệ nguồn hình sao, sử dụng T-SQL (vd: MicroStrategy OLAP Services)
- MOLAP** - multidimensional OLAP: OLAP sử dụng csdl đa chiều (ví dụ Cognos BI 8 Analysis, ProClarity Analytics 6)
- HOLAP**: hỗ trợ cả ROLAP và MOLAP
  - SQL server analysis hỗ trợ ROLAP, MOLAP và HOLAP

23

## 2. Khai thác dữ liệu đa chiều

Nguồn: <http://www.tutorialspoint.com/dwh/>

24

## OLAP vs Reporting

### Report

- Ưu điểm:**
  - Dễ tạo, dễ quản lý, dễ sử dụng
  - Thường sử dụng report trong BI khi các yêu cầu trình bày định dạng là hoàn toàn đơn giản và tĩnh (static)
- Khuyết điểm:**
  - Không linh động
  - Không tương tác
  - Nếu người dùng muốn thay đổi nội dung báo cáo hoặc muốn xem dữ liệu tại một mức cao hơn hoặc thấp hơn thì cần phải thiết kế lại report.

26

## Các phép toán

### 1. Xoay trục (pivot/rotate)

- Thay đổi trực ngang thành dọc và ngược lại
- Chọn 2 trên n trực để biểu diễn chính

### 2. Tổng hợp (roll-up)

- Tổng hợp/ kết hợp các ô bên trong 1 chiều thường là phép tổng, đếm
- Sử dụng định nghĩa phân cấp chiều

27

## Các phép toán

### 3. Chi tiết (drill-down)

- Cung cấp dữ kiện theo mức chi tiết của chiều

### 4. Cắt lát (slice and dice)

- Slide:** Chọn 1 tập con của khối dựa vào ràng buộc giá trị của 1 số chiều (ví dụ: chọn các ô có tháng = T1 theo chiều thời gian)
- Dice:** Chọn 1 tập con của khối dựa vào ràng buộc của 2 hoặc nhiều chiều

28

## Roll-up

**Roll-up**

- chiều cửa hàng  
→ Khu vực

Cửa hàng là chiều con của KhuVuc

29

## Roll-up

30

## Roll-up

32

## Drill-down

**Drill-down**

- chiều thời gian Tháng → ngày

Tháng là cấp cha của ngày

33

## Slice

35

	Sản phẩm	TPHCM	HN	DN
Bánh mì	30	26	11	
Phô mai	15	35	5	
Sữa chua	20	18	38	

### Slice - cắt lát

Biểu diễn dữ liệu 3 chiều, giống như 1 CUBE

- Sự kiện sản phẩm P1 được bán tại thời điểm T1, tại cửa hàng C1

### Dice - trích khối con

### Bài tập

- Xác định các thao tác OLAP đáp ứng các nhu cầu phân tích sau:
  - Lập báo cáo số lượng sản phẩm P1 được bán ra trong tháng 1,2 tại tất cả cửa hàng
  - Lập thống kê số lượng sản phẩm bán ra tại cửa hàng C3 qua mỗi tháng
  - Lập thống kê số lượng sản phẩm bán ra tại cửa hàng C1, C2 qua mỗi tháng
- Ví dụ dữ liệu mẫu cho từng yêu cầu trên

### Tham khảo

- Josée Ranger, Lacroix, SAS Institute (Canada) Inc. *Introduction to SAS OLAP: A Solution for the Curious and Impatient*. OASUS: FALL 2008
- V. Estivill-Castro, J. Gasston. *Adv Topics in Computing Science Knowledge Discovery and Data Mining*
- Vincent Rainardi. *Building a Data Warehouse With Examples in SQL Server*

### Bài tập

Tìm hiểu ưu khuyết của ROLAP, MOLAP, HOLAP

### Tham khảo

- <http://www.1keydata.com/datawarehousing/molap-rolap.html>

**MDX**

Multidimensional Expressions

Giảng viên: Hồ Thị Hoàng Vy  
2013, Jan 27

1

## Nội dung

- ❑ Khái niệm MDX
- ❑ Cú pháp MDX
- ❑ Một số MDX function
- ❑ Một vài ứng dụng

2

## Khái niệm

- **MDX:** Multi-Dimensional eXpressions
  - SQL dùng để truy vấn csdl quan hệ, MDX được dùng để truy vấn csdl đa chiều
  - Ngôn ngữ chuẩn, được định nghĩa bởi MS → truy vấn OLAP server
  - Dùng để tính toán và phân tích trên cấu trúc khôi OLAP

3

## Khái niệm

- Mỗi liên hệ của dữ liệu trong CUBE được chia thành các quan hệ sau:

Dimensions  
Hierarchies  
Levels  
Members

4

## Khái niệm

- **Dimensions and Members**
  - Một dimension có thể gồm nhiều levels
  - Mỗi levels bao gồm một số các members
- **Axes**
  - Đè cập tới dimension khi truy vấn Cube
  - 1 axes có thể kết hợp nhiều dimension của Cube
- **Measures:**
  - Thuộc tính (gồm giá trị số) được thống kê và phân tích
- **Default Member :** ALL (mức top)
- **Default measure:** measure đầu tiên mô tả trong CUBE
- **Tuple:** một lát dữ liệu trong Cube
- **SET:** một tập hợp gồm zero, một hoặc nhiều tuples

5

## Ví dụ

Chiều SanPham

```

graph TD
    ALL_SanPham[ALL SanPham] --> LoaiSP[LoạiSP]
    LoaiSP --> TenSP[TenSP]
    
```

Members của chiều sản phẩm

```

graph TD
    SanPham_ALL[Sản phẩm ALL] --> GiaiKhalt[Giải khát]
    SanPham_ALL --> ThucAn[Thực ăn]
    GiaiKhalt --> Coca[Coca]
    GiaiKhalt --> Pepsi[pepsi]
    ThucAn --> BanhMi[Bánh mì]
    ThucAn --> PhoMai[Phô mai]
    PhoMai --> Bo[Bơ]
    
```

Biểu thức truy vấn trả về tập các members:

1. [SanPham].[LoaiSP].Pepsi
2. [SanPham].[TenSP].Members = { Coca, pepsi, bánh mì, phô mai, bơ }
3. [SanPham].[Giải khát].CHILDREN = {Coca, pepsi}
4. [SanPham].[Thực ăn].[Phô mai].[Bơ] = {phô mai, bơ}
5. DESCENDANTS[[SanPham].[Giải khát], TenSP] = {coca, pepsi}

6

## Cú pháp MDX

```
[ WITH MEMBER <member_name> AS <value_expr> |
SET <set_name> AS <set_expr>
SELECT <axis_specification> [, <axis_specification>, ...]
FROM <cube_specification>
[ WHERE <slicer_specification>]
```

**Giải thích**

- Select: xác định các trục chiều thành viên
- From: khối được truy vấn được đặt tên trong mệnh đề From.
- Where: giới hạn dữ liệu kết quả. Thường đề cập tới việc cắt lát (slicer)

7

## Cú pháp MDX

**MDXQuery2.mdx - APCUBE (y-Poly)**

```
select [Measures] on columns
from [test]
```

**QVý**

```
select measures.members on columns
from [test]
```

8

## Cú pháp MDX

- Ví dụ

```
select [Measures].[Soluong] on columns
from [Test]
```

9

## Cú pháp MDX

- Ví dụ:

```
select [Ngay].[Date].[Nam] on columns
from [Test]
```

- Phân tích:
  - [Ngay].[Date].[Nam] là thuộc định dạng: **[Dimension].[Hierarchy].[Level]**

10

## Cú pháp MDX

- Ví dụ

```
Select [Measures].[GiaTien] on columns,
[Cuhang].[Tench].[Tench] on rows
From [Test]
```

	Giatien
dealot	(null)
jade	6000
jupiter	6500
vnshop	500

- Mệnh đề from chỉ nêu tên của 1 cube

11

## Cú pháp MDX

- Thêm 1 cột tổng vào cuối

```
Select { [Ngay].[Date].[Nam], [Ngay].[Date] } on columns
From [Test]
```

**Ghi chú:**

Nếu có nhiều cột, cách nhau bởi các dấu phẩy ','  
Dấu ngoặc nhọn {} bao bên ngoài là bắt buộc  
[Ngay].[Date] là định dạng: [Dimension].[Hierarchy]

12

## Cú pháp MDX

- Ví dụ

```
select { [Ngay].[Date].[Nam], [Ngay].[Date] } on columns,
       [Sanpham].[Hierarchy].[Maloai] on rows
from [Test]
```

	N1	N2	N3	Unknown	All
L1	10	52	(null)	(null)	62
L2	(null)	100	(null)	(null)	100
L3	(null)	20	(null)	(null)	20
Unknown	(null)	(null)	(null)	(null)	(null)
All	(null)	(null)	(null)	(null)	(null)

13

## Cú pháp MDX

```
select
    { [Ngay].[Date].[Nam], [Ngay].[Date] } on columns,
    { [Sanpham].[Hierarchy].[Maloai], [Sanpham].[Hierarchy] } on rows
From [Test]
```

Messages Results

	N1	N2	N3	Unknown	All
L1	10	52	(null)	(null)	62
L2	(null)	100	(null)	(null)	100
L3	(null)	20	(null)	(null)	20
Unknown	(null)	(null)	(null)	(null)	(null)
All	10	172	(null)	(null)	182

Select non empty

```
{ [Ngay].[Date].[Nam], [Ngay].[Date] } on columns,
{ [Sanpham].[Hierarchy].[Maloai], [Sanpham].[Hierarchy] } on rows
From [Test]
```

14

## Cú pháp MDX

```
Select non empty { [Ngay].[Date].[Nam], [Ngay].[Date] } on columns,
Non empty { [Sanpham].[Hierarchy].[Maloai], [Sanpham].[Hierarchy] } on rows
From [Test]
Where [Measures].[SoLuong]
```

	N1	N2	N3	Unknown	All
L1	10	52	(null)	(null)	62
L2	(null)	100	(null)	(null)	100
L3	(null)	20	(null)	(null)	20
Unknown	(null)	(null)	(null)	(null)	(null)
All	10	172	(null)	(null)	182

15

## Cú pháp MDX

- Câu hỏi

```
select
    [Measures].[Soluong].[Measures].[GiaTien]
    on columns,
    [KhachHang].[Dia chi].members on rows
From [Test]
```

Lỗi

16

## Cú pháp MDX

- Sửa lỗi

```
Select {
    [Measures].[Soluong],[Measures].[GiaTien]
    on columns,
    [Khach Hang].[Dia chi].members on rows
}
From [Test]
```

	Soluong	GiaTien
All	182	13000
DBP	45	3000
PN	2	500
PXL	75	4500
Tan binh	60	5000

17

## Cú pháp MDX

Cho phân cấp chiều Ngày như sau: Dimension.Hierarchy.Level

Dimension Structure

Hierarchy

Attributes

Data Source View

Server: SQL2008 Database: Analysis Services

18

## Cú pháp MDX

- Hãy viết các truy vấn sau:
  - Cho biết số lượng sản phẩm bán ra tại các cửa hàng trong hai năm N1, N2
  - Cho biết số lượng sản phẩm bán ra tại các cửa hàng qua mỗi năm

19

## Cú pháp MDX

```
SELECT { [Ngay].[Date].[Nam].&[N1],
          [Sanpham].[Mau].[Mau] } ON COLUMNS,
          { [Cuahang].[Mac].[Members} ON ROWS
FROM [test]
WHERE ([Measures].[Soluong])
```

21

X

- Executing the query ...
  - Query (1, 8) Members, tuples or sets must use the same hierarchies in the function.*
- Execution complete

## Cú pháp MDX

- level.MEMBERS**

```
Select { [Measures].[GiaTien],[Measures].[SoLuong} on columns,
          [Khachhang].[Makh].members on rows
From [Test]
```

- Member là level thấp nhất trong phân cấp
- .member:** trả về tất cả all members của 1 chiều cho trước, hierarchy or level

22

## Cú pháp MDX

- Member**
  - Có thể được tham chiếu bởi tên hoặc key
  - Member name và member key phân biệt bởi dấu &
  - Ví dụ
    - [Time].[2nd half].&[Q4]
    - [Time].[2nd half].[4th quarter]
  - Dấu ] dùng để tránh kí tự ] trong tên :
    - [Premier [150]] 98]

23

## Một số function MDX

- Member functions: MDX cung cấp nhiều functions để rút trích các member từ các level, sets, tuple.
  - FirstChild**
  - Ví dụ**

```
SELECT [Date].[Calendar Year].[CY 2001] on 0
FROM [Adventure Works]
```

Hoặc

```
SELECT [Date].[Calendar Year].FirstChild on 0
FROM [Adventure Works]
```

24

## Một số function MDX

- Member functions (tt)
  - Một số function khác:
    - LastChild
    - LastSibling
    - Parent
    - Ancestor
    - ....
  - Tham khảo: <http://msdn.microsoft.com/en-us/library/ms145970.aspx>

25

## Một số function MDX

- With keyword – Calculated member**
  - Member có thể được tạo và trả về dữ liệu được tính toán thay vì lưu trữ dữ liệu trong khối để truy vấn
  - Những member này được gọi là calculated member
  - Cú pháp:  
**WITH MEMBER parent.name AS 'expression'**
- Ví dụ:**  
WITH MEMBER [Measures].[PackagesForecast] AS  
'[Measures].[Packages] \* 1.1'

26

## Một số function MDX

```
WITH MEMBER [Time].[1997].[H1] AS
[Time].[1997].[Q1] + [Time].[1997].[Q2]
MEMBER [Time].[1997].[H2] AS
[Time].[1997].[Q3] + [Time].[1997].[Q4]

SELECT {[Time].[1997].[H1], [Time].[1997].[H2]} ON COLUMNS
[Store].[Store Name].MEMBERS ON ROWS
FROM [Sales]
WHERE (Measures.[Profit])
```

27

## Một số function MDX

- ORDER**
  - Cú pháp: ORDER(set, expression, [, ASC | DESC [...]])
  - Ví dụ:** liệt kê tất cả measure cho mỗi khách hàng theo thứ tự giảm dần về số lượng mua

```
SELECT Measures.MEMBERS ON COLUMNS,
ORDER ([Sanpham].[Hierarchy].[Maloai].members
, Measures.[SoLuong], DESC ) ON ROWS
FROM [Test]
```

	Soluong	Giatien	Fact Count
L2	100	5500	3
L1	62	5500	3
L3	20	2000	2
Unknown	(null)	(null)	(null)

29

## Một số function MDX

*Nếu chỉ quan tâm 2 loại sản phẩm L1 & L2:*

```
SELECT Measures.MEMBERS ON COLUMNS,
ORDER ([Sanpham].[Hierarchy].[Maloai].[L1]:[L2], Measures.[SoLuong],
DESC ) ON ROWS
FROM [Test]
```

	Soluong	Giatien	Fact Count
L2	100	5500	3
L1	62	5500	3

30

## Một số function MDX

- HEAD - TOP COUNT**
  - Chỉ xuất ra 3 loại sản phẩm có số lượng bán ra nhiều nhất

```
SELECT Measures.MEMBERS ON COLUMNS,
HEAD ( ORDER ([Sanpham].[Hierarchy].[Maloai].members
, Measures.[SoLuong], DESC ), 3) ON ROWS
FROM [Test]
```

	Soluong	Giatien	Fact Count
L2	100	5500	3
L1	62	5500	3
L3	20	2000	2

31

## Một số function MDX

- Case...when**

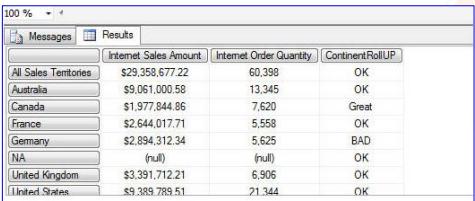
```
WITH MEMBER [Measures].ContinentRollUP AS
CASE [Measures].[Internet Order Quantity]
WHEN 7620 THEN 'Great'
WHEN 5625 THEN 'BAD'
ELSE 'OK'
END

SELECT
{[Measures].[Internet Sales Amount], [Measures].[Internet Order Quantity]
, [Measures].ContinentRollUP } ON COLUMNS
,[Sales Territory].[Sales Territory Country].Members ON ROWS
FROM [Adventure Works]
```

32

## Một số function MDX

- Kết quả



33

## Một số function MDX

**Tất cả kết quả đều thể hiện 2D →  
Vậy với 3D???**

34

## Một số function MDX

- Crossjoin (set 1, set 2)
  - Ví dụ: báo cáo số lượng mỗi loại sản phẩm bán ra tại mỗi cửa hàng trong năm N1

```
select crossjoin ([Sanpham].[Hierarchy].[Maloai].Members, [Cuahang].[Tench].Members) on rows,
[Ngay].[Manam].&[N1] on columns
from test
```



35

## Slice dimension vs Filter

- SLICE ON PRODUCT DIM

```
SELECT {[Time].[Year].[1997].CHILDREN} ON COLUMNS,
{[Store].[Store City].MEMBERS} ON ROWS
FROM [Sales]
WHERE {[Product].[Product Family].[Drink], [Measures].[Store Sales]}
```

- FILTER(set, search condition) function.

```
SELECT {[Time].[Year].[1997].CHILDREN} ON COLUMNS,
FILTER {[Store].[Store City].MEMBERS}, {Measures.[Unit Sales]}, [Time].[1997]) > 1000 ) ON ROWS
FROM [Sales]
WHERE {[Measures].[Store Sales]}
```

36

## MDX vs SQL

	MDX	SQL
Select	Định nghĩa nhiều trục chiều	Mô tả column layout
From	chỉ chứa tên 1 cube	[1-n] Table name
Where	mô tả trục lát cắt dữ liệu	Điều kiện filter dữ liệu

Lưu ý:  
Không thể yêu cầu 1 lát cắt cho nhiều members của cùng 1 chiều  
Calculated measure:  
<http://www.sqlcircuit.com/2013/01/how-to-create-calculated-members-in-ssas.html>

37

## SQL mở rộng - ROLLUP

```
SELECT dept, job, count(*), sum(salary)
FROM employees
GROUP BY ROLLUP(dept, job)
```

dept.	name	job	salary
10	Alex	manager	2450
10	Jack	president	5000
20	Jill	manager	2975
10	Ann	clerk	1300
20	Mike	clerk	900
20	Mary	clerk	1000
20	Matt	analyst	2800
20	Sarah	analyst	3200

```
dept job count(*) sum(salary)
----- -----
10 CLERK 1 1300
10 MANAGER 1 2450
10 PRESIDENT 1 5000
10 ANALYST 3 8750
20 CLERK 2 1900
20 MANAGER 1 2975
20 ANALYST 5 10875
20 8 19625
```

TK: Dr Goran Nenadic – School of Informatics , University of Manchester

43

## SQL mở rộng - CUBE

**Table:**

dept.	name	job	salary
10	Alex	manager	2450
10	Jack	president	5000
20	Jill	manager	2975
10	Ann	clerk	1300
20	Mike	clerk	900
20	Mary	clerk	1000
20	Matt	analyst	2800
20	Sarah	analyst	3200

**SQL:**

```
SELECT
    dept, job, count(*), sum(salary)
FROM employees
GROUP BY CUBE(dept, job)
```

**Output:**

dept	job	count(*)	sum(salary)
10	CLERK	1	1300
10	MANAGER	1	2450
10	PRESIDENT	1	5000
10		3	8750
20	ANALYST	2	6000
20	CLERK	2	1900
20	MANAGER	1	2975
20		5	10875
	ANALYST	2	6000
	CLERK	3	3200
	MANAGER	2	5425
	PRESIDENT	1	5000
		8	19625

44

## Grouping set

- Allows to specify particular groupings
- Total shipment quantities by supplier and by product

**SQL:**

```
SELECT
    S#, P#, SUM(QTY)
FROM SP
GROUP BY
    GROUPING SETS ( S#, P# )
```

**Output:**

S#	P#	Tot.
S1	null	500
S2	null	200
null	P1	300
null	P2	150
null	P3	250

45

## Grouping set

- roll-up along a given dimension

**SQL:**

```
SELECT
    S#, P#, SUM(QTY)
FROM SP
GROUP BY
    ROLLUP ( S#, P# )
```

**Output:**

S#	P#	Tot.
S1	P1	200
S1	P2	100
S1	P3	200
S2	P1	100
S2	P2	50
S2	P3	50
S1	null	500
S2	null	200
null	null	700

**Diagram:**

1) aggregate with the finest granularity (GROUP BY S#, P#)  
2) then with the next level of granularity (GROUP BY S#)  
3) then the grand total (with no GROUP BY clause)

46

## Ví dụ:

**SQL:**

```
SELECT model, year, color, sum(sales) as sales
FROM sales
WHERE model in ('Chevy') AND year BETWEEN 1990 AND 1992
GROUP BY CUBE (model, year, color);
```

**Output:**

Chevy Sales Cross Tab			
Chevy	1990	1991	1992
black	50	85	154
white	40	115	199
<b>Total</b>	<b>90</b>	<b>200</b>	<b>353</b>
<b>(ALL)</b>			<b>1286</b>

47

## Ví dụ:

**SQL:**

```
SELECT model, year, color, sum(sales) as sales
FROM sales
WHERE model in ('Chevy')
AND year BETWEEN 1990 AND 1992
GROUP BY CUBE (model, year, color);
```

**Gồm 8 grouping:**

- (model, year, color),
- (model, year),
- (model, color),
- (year, color),
- (model),
- (year),
- (color),
- ()

48

## Kết quả

**DATA CUBE:**

Model	Year	Color	Sales
CHEVY	1990	red	5
CHEVY	1990	white	87
CHEVY	1990	blue	62
CHEVY	1991	red	64
CHEVY	1991	white	95
CHEVY	1991	blue	49
CHEVY	1992	red	31
CHEVY	1992	white	54
CHEVY	1992	blue	71
FORD	1990	red	64
FORD	1990	white	62
FORD	1990	blue	53
FORD	1991	red	62
FORD	1991	white	9
FORD	1992	red	27
FORD	1992	white	62
FORD	1992	blue	39

**CUBE:**

SALES			
Model	Year	Color	Sales
CHEVY	1990	red	5
CHEVY	1990	white	87
CHEVY	1990	blue	62
CHEVY	1991	red	64
CHEVY	1991	white	95
CHEVY	1991	blue	49
CHEVY	1992	red	31
CHEVY	1992	white	54
CHEVY	1992	blue	71
FORD	1990	red	64
FORD	1990	white	62
FORD	1990	blue	53
FORD	1991	red	62
FORD	1991	white	9
FORD	1992	red	27
FORD	1992	white	62
FORD	1992	blue	39

49

## ROLUP - ROLAP

```
SELECT model, year, color, sum(sales) as sales
FROM   sales
WHERE  model in ('Chevy') AND year BETWEEN 1990 AND 1992
GROUP BY ROLLUP (model, year, color);
```

- Chi thực hiện grouping 4 nhóm sau:
  - (model, year, color),
  - (model, year),
  - (model),
  - ()

50

## VD2

- CREATE TABLE tblPopulation (
 Country VARCHAR(100),
 [State] VARCHAR(100),
 City VARCHAR(100),
 [Population (in Millions)] INT
 GO
- INSERT INTO tblPopulation VALUES('India', 'Delhi','East Delhi',9 )
 INSERT INTO tblPopulation VALUES('India', 'Delhi','South Delhi',8 )
 INSERT INTO tblPopulation VALUES('India', 'Delhi','North Delhi',5.5)
 INSERT INTO tblPopulation VALUES('India', 'Delhi','West Delhi',7.5)
 INSERT INTO tblPopulation VALUES('India', 'Karnataka','Bangalore',9.5)
 INSERT INTO tblPopulation VALUES('India', 'Karnataka','Belur',2.5)
 INSERT INTO tblPopulation VALUES('India', 'Karnataka','Manipal',1.5)
 INSERT INTO tblPopulation VALUES('India', 'Maharashtra','Mumbai',30)
 INSERT INTO tblPopulation VALUES('India', 'Maharashtra','Pune',20)
 INSERT INTO tblPopulation VALUES('India', 'Maharashtra','Nagpur',11 )
 INSERT INTO tblPopulation VALUES('India', 'Maharashtra','Nashik',6.5)
 GO

51

## Ví dụ

```
SELECT Country,[State],City,
SUM ([Population (in Millions)]) AS [Population (in Millions)]
FROM tblPopulation
GROUP BY Country,[State],City WITH ROLLUP
```

Country	State	City	Population (in Millions)
1 India	Delhi	East Delhi	9
2 India	Delhi	South Delhi	8
3 India	Delhi	North Delhi	5
4 India	Delhi	West Delhi	7
5 India	Karnataka	Bangalore	9
6 India	Karnataka	Belur	2
7 India	Karnataka	Manipal	1
9 India	Karnataka	NULL	12 Karnataka Total
10 India	Maharashtra	Mumbai	30
11 India	Maharashtra	Nagpur	11
12 India	Maharashtra	Nashik	6
13 India	Maharashtra	Pune	20
14 India	Maharashtra	NULL	67 Maharashtra Total
15 India	NULL	NULL	108
16 NULL	NULL	NULL	108 India Total

52

Country	State	City	Population (in Millions)
1 India	Delhi	East Delhi	9
2 India	Delhi	North Delhi	5
3 India	Delhi	South Delhi	8
4 India	Delhi	West Delhi	7
5 India	Delhi	NULL	29 Delhi Total
6 India	Karnataka	Bangalore	9
7 India	Karnataka	Belur	2
8 India	Karnataka	Manipal	1
9 India	Karnataka	NULL	12 Karnataka Total
10 India	Maharashtra	Mumbai	30
11 India	Maharashtra	Nagpur	11
12 India	Maharashtra	Nashik	6
13 India	Maharashtra	Pune	20
14 India	Maharashtra	NULL	67 Maharashtra Total
15 India	NULL	NULL	108
16 NULL	NULL	NULL	108 India Total

<http://blog.sqlauthority.com/2010/02/24/sql-server-introduction-to-rollup-clause/>

53

## Tham khảo

- Josée Ranger, Lacroix. SAS Institute (Canada) Inc. *Introduction to SAS OLAP: A Solution for the Curious and Impatient*. OASUS: FALL 2008
- V. Estivill-Castro, J. Gasston. *Adv Topics in Computing Science Knowledge Discovery and Data Mining*
- Vincent Rainardi. *Building a Data Warehouse With Examples in SQL Server*

55

Thank you

Giảng viên: Hồ Thị Hoàng Vy  
2013, Jan 27

56

## Bài tập

Tìm hiểu ưu khuyết của ROLAP, MOLAP, HOLAP

**Tham khảo**

- <http://www.1keydata.com/datawarehousing/molap-rolap.html>

57



**BI application**

Giảng viên: Hồ Thị Hoàng Vy  
2013, Jan 27

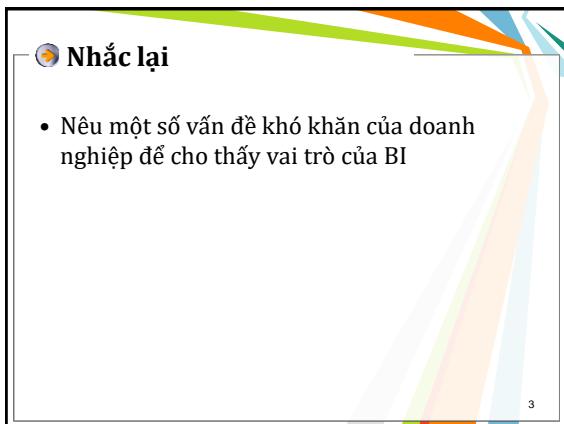
1



## Nội dung

- Adhoc query
- Basic report
- Business analysis (OLAP)
- Mining
- Dashboard

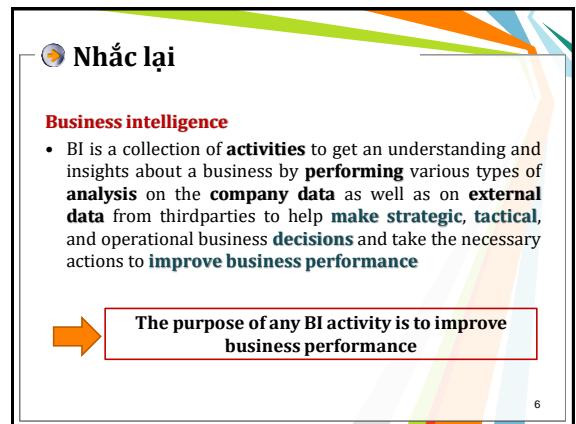
2



## Nhắc lại

- Nêu một số vấn đề khó khăn của doanh nghiệp để cho thấy vai trò của BI

3



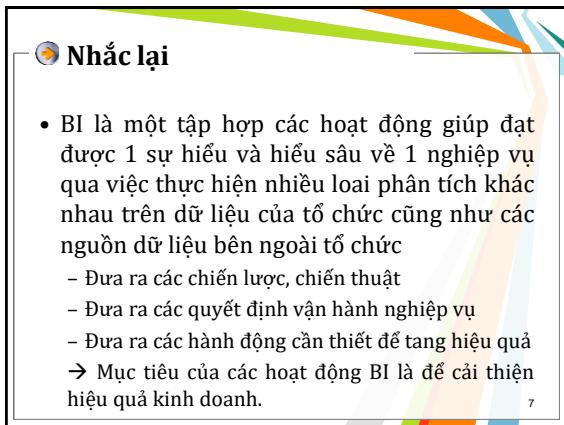
## Nhắc lại

### Business intelligence

- BI is a collection of **activities** to get an understanding and insights about a business by **performing** various types of **analysis** on the **company data** as well as on **external data** from thirdparties to help **make strategic, tactical, and operational business decisions** and take the necessary actions to **improve business performance**

The purpose of any BI activity is to improve business performance

6



## Nhắc lại

- BI là một tập hợp các hoạt động giúp đạt được 1 sự hiểu và hiểu sâu về 1 nghiệp vụ qua việc thực hiện nhiều loại phân tích khác nhau trên dữ liệu của tổ chức cũng như các nguồn dữ liệu bên ngoài tổ chức
  - Đưa ra các chiến lược, chiến thuật
  - Đưa ra các quyết định vận hành nghiệp vụ
  - Đưa ra các hành động cần thiết để tăng hiệu quả

→ Mục tiêu của các hoạt động BI là để cải thiện hiệu quả kinh doanh.

7



## Nhắc lại

- Ví dụ:
  - analyzing customer profitability
  - studying product profitability
  - evaluating sales figures across different products and regions
  - exploring accounts profitability
  - examining supplier performance
  - discovering customer risk patterns

8

## Nhắc lại

- Các ứng dụng BI có thể truy cập dữ liệu từ:
  - Các hệ thống ODS
  - Các hệ thống ERP, CRM...
  - Kho dữ liệu

9

## Phân loại ứng dụng BI

- Kho dữ liệu được tạo để phục vụ cho BI
- Nếu 1 KDL được xây dựng mà không rõ là phục vụ cho loại BI nào tổ chức cần → không tạo được giá trị gì.

### Các ứng dụng BI truy vấn dữ liệu từ KDL

- Đừng tập trung vào sẽ dùng tool gì
- Hãy tập trung tìm ra các loại câu hỏi mà người dùng sẽ hỏi để xác định nội dung của KDL, loại báo biểu cần tạo và mục đích tạo để làm gì...

10

## Phân loại ứng dụng BI

Table 8-1 Business Intelligence Categories		
Type	Information You Want	See This Chapter
Basic querying and reporting	"Tell me what happened."	9
Business analysis (OLAP)	"Tell me what happened and why."	10
Data mining	"Tell me what might happen" or "Tell me something interesting."	11
Dashboards and scorecards	"Tell me a lot of things, but don't make me work too hard."	12

Data warehouse for Dummies 2<sup>nd</sup> - Thomas C.Hammergren & Alan R.Simon

11

## Phân loại ứng dụng BI

This is an example of a data warehouse system  
ETL  
Front end application  
Building DW with SQL server

12

## BI Report

- Định dạng bảng tính, pivot (group, sort, filter), hoặc các biểu đồ (line chart, bar chart, pie chart, ...)
- Có thể thêm các parameter để truyền động điều kiện
- Có thể cài đặt để gửi report tới người dùng cách tự động qua mail (hằng ngày, hằng tuần...)

- Ưu:**
  - Dễ tạo, dễ quản lý, dễ dùng
  - Sử dụng report trong BI khi định dạng của yêu cầu đơn giản và tĩnh
- Khuyết:**
  - Không linh động, không tương tác
  - Muốn xem dữ liệu ở các phân cấp khác nhau → thiết kế lại report

13

## Các loại biểu đồ

	Loại biểu đồ
<b>So sánh, xếp hạng</b>	<ul style="list-style-type: none"> <li>Biểu đồ cột, cột chồng (theo chiều ngang lần dọc)</li> <li>Dữ liệu được thể hiện bởi các điểm</li> </ul>
<b>Điển biến theo thời gian, độ chênh lệch của các giá trị</b>	<ul style="list-style-type: none"> <li>Biểu đồ cột (so sánh giữa các giá trị)</li> <li>Dữ liệu thể hiện bởi các điểm nối với nhau bằng các đoạn thẳng (nhấn mạnh sự thay đổi dữ liệu)</li> </ul>
<b>Tìm mối tương quan</b>	<ul style="list-style-type: none"> <li>Dữ liệu được thể hiện bởi từng điểm và có trend line</li> <li>Đồ thị</li> <li>Biểu đồ cột</li> </ul>
<b>So sánh tì trọng</b>	<ul style="list-style-type: none"> <li>Biểu đồ tròn</li> <li>Biểu đồ cột, cột chồng</li> </ul>

14

## BI Analytic

- Cung cấp gợi ý cho quá trình ra quyết định qua việc truy cập dữ liệu từ nhiều nguồn
- Truy cập dữ liệu đa chiều linh hoạt, còn gọi là **OLAP app**.
- Người dùng có thể rollup, drilldown, slice & dice dữ liệu
  - Phân tích lợi nhuận từ sản phẩm và khách hàng
  - Tìm hiểu làm thế nào để giảm chi phí tồn...
- Một số ứng dụng analytic (MOLAP app) tích hợp sở hữu cả HQTCSDL đa chiều như: SSAS, SAP business warehouse, Cognos 8 BI analytic....
- Ứng dụng analytic (ROLAP app): MicroStrategy

16

## BI Analytic

- Ưu**
  - Linh hoạt
  - Truy vấn adhoc, phân tích dữ liệu từ nhiều cube....
- giúp phân tích chuyên sâu, có cái nhìn tổng quan về hiệu suất kinh doanh hiện tại
- Khuyết**
  - Phức tạp
  - Người dùng phải mất thời gian làm quen với tool

17

## BI Data mining

- Khám phá dữ liệu trong kho để tìm mẫu và các mối liên hệ tiềm ẩn của dữ liệu
  - Trong bán hàng: Tìm mối liên hệ giữa người mua và sản phẩm mua
  - Trong tài chính: dự đoán khả năng và sự sẵn sàng trả nợ dựa vào hành vi thanh toán, mức nợ hiện tại, lịch sử thẻ tín dụng...
  - Trong quản lý mối liên hệ khách hàng:

18

## BI Data mining

- Recommender system
  - [In retail](#)

19

## BI Dashboard

- Biểu diễn dữ liệu thống kê cấp cao theo các biểu đồ đồ họa (graphical gadgets, gauges, charts, indicators, and color-coded maps...)
- Dữ liệu dựa vào KDL, được cập nhật hàng ngày, hoặc hàng tuần, tháng phụ thuộc vào nhu cầu cần suât đo lường hiệu quả kinh doanh
- Nhu cầu “real time dashboard” ngày càng phổ biến, do đó truy vấn dữ liệu từ ODS thay vì KDL

20

## Safety Incident Dashboard: Tableau

21

### Safety Incident Dashboard: Excel in Web Browser

22

### BI ALERT

- Cảnh báo cho người dùng khi có 1 sự kiện hoặc một điều kiện nào đó xảy ra, ví dụ:
  - Trong bán hàng, cấu hình tỷ lệ % doanh số bán hàng tại các store
  - Khi số lần đóng tài khoản trên toàn quốc vượt quá số lần cho phép
  - Khi lợi nhuận của một số loại sản phẩm trên toàn vùng thấp hơn một giới hạn đặt trước.
  - ....
- Cân phân biệt giữa cảnh báo mức KDL và cảnh báo mức hệ thống tác vụ
- Cảnh báo sớm, giúp ngăn ngừa tình trạng xấu xảy ra. Tuy nhiên, chỉ làm được với dữ liệu đơn giản

23

### BI portal

- Là các ứng dụng đóng vai trò “gateway” giúp truy cập và quản lý các ứng dụng BI reports, analytics, data mining và dashboard
- Giúp các ứng dụng BI có thể sẵn sàng ở 1 nơi trung tâm
- Quản lý bảo mật tập trung (chỉ cần 1 lần đăng nhập)

24

### BI portals

25

### Lợi ích

*"The time savings are enormous, and we can now make business decisions more readily."*

- Mike Dayton, Plant Manager at Prest-O-Fit

- Nhanh
- Dễ dùng
- Chính xác hơn
- Chi tiết hơn
- Hỗ trợ quyết định

26

### Data visualization

- Vấn đề
  - Làm sao chuyển đạt dữ liệu đến người dùng hiệu quả?
  - Thể hiện dữ liệu (data visualization) giúp truyền thông dữ liệu hiệu quả thông qua biểu diễn đồ họa
  - Ví dụ:
    - Các báo cáo
    - Quản lý các hoạt động kinh doanh
    - Khám phá mối tương quan dữ liệu <> dữ liệu thô

27

Business Intelligence Consulting Services		
Data Visualization Consulting	BI and Analytics	Microsoft and BI
Data Visualization Services	Big Data Analytics	SQL Server Analysis Services
Analytical Dashboards	Data Mining	What is PowerPivot?
Trend Analysis	Data Cube Optimization	BI with Excel
Sparklines	Interactive Data Visualization	Excel and Analysis Services
Outlier Detection	Qlikview Consulting	Excel to Smart Client Conversion
Cluster Analysis	Spotfire Consulting	MS Excel Web Applications
What is an Outlier?	Tableau Consulting	Data Warehouse with SQL Server

<http://www.practicaldb.com/business-intelligence-analytics-2/>  
<http://www.practicaldb.com/demos/>

**PRACTICAL's Clients: Location & Service**

**Map of Clients**  
Sized by Number of Users, Colored by Type of Service

State	Clients
CA	1000
NY	800
IL	600
PA	500
WA	400
TX	300
VA	250
GA	200
MD	180
MI	150
NC	120
DE	100
RI	80
CT	70
AK	60
IN	50
ME	40
SD	30
WV	20
NE	15
MT	10
HI	5
MS	2
WY	1

**Service**

- Data Analytics and BI
- Data Visualization
- Machine Learning
- Database Migration
- Database Optimization
- Prototyping
- Smart Client
- SQL Audit
- System Tuning
- Service Testing

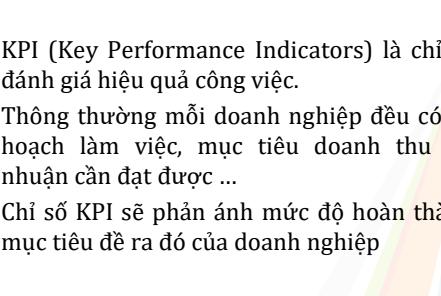
**PCAs Services**

- ✓ Data Analytics and BI
- ✓ Data Visualization
- ✓ Machine Learning
- ✓ Database Migration
- ✓ Database Optimization
- ✓ Prototyping
- ✓ Smart Client
- ✓ SQL Audit
- ✓ System Tuning

**Note:** Data is shown for City, State and ZIP. The data is presented as ZIP codes which ranges from 2 to 217 ZIP codes based on State and Service.

About Tableau: [www.tableau.com/tableau](http://tableau.com/tableau)

User Count: 2,371



- KPI (Key Performance Indicators) là chỉ số đánh giá hiệu quả công việc.
- Thông thường mỗi doanh nghiệp đều có kế hoạch làm việc, mục tiêu doanh thu lợi nhuận cần đạt được ...
- Chỉ số KPI sẽ phản ánh mức độ hoàn thành mục tiêu đề ra đó của doanh nghiệp

## Phân tích kết hợp với các loại biểu đồ (Data visualization)