Data Warehousing e Tecnologia OLAP para Data Mining

- O que é um data warehouse?
- O modelo de dados multi-dimensional
- Arquitectura de data warehouses
- Implementação de data warehouses
- Mais aspectos da tecnologia multi-dimensional
- De data warehousing a data mining

O que é um Data Warehouse?

- Definido de várias maneiras diferentes, mas não de uma forma rigorosa.
  - Uma base dados de suporte a decisão que é mantida separadamente da base operacional da organização.
  - Suporta processamento de informação fornecendo uma plataforma sólida para análise de dados históricos, consolidados.
- "A data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision-making process."—W. H. Inmon
- Data warehousing:
  - O processo de construir e usar data warehouses

Data Warehouse-Orientado por Tema

- Organizado em torno de temas importantes, tais como cliente, produto, vendas.
- Focado na modelação e análise de dados para quem toma decisões, em vez de operações diárias e processamento de transacções.
- Fornece uma visão simples e concisa sobre questões de um tema particular através da exclusão de dados que não são importantes no suporte ao processo de decisão.

Data Warehouse—Integrado

- Construído por integração de múltiplas e heterogéneas fontes de dados
  - Bases de dados relacionais, ficheiros simples, registos de transacções on-line
  - São aplicadas técnicas de limpeza de dados e integração de dados.
  - É assegurada a consistência na convenção de nomes, codificação de estruturas, atributos de medidas, etc. entre diferentes fontes de dados
    - E.g., Hotel price: currency, tax, breakfast covered, etc.
  - Quando a informação é movida para o warehouse, é feita a conversão.

Data Warehouse—Variável Tempo

- O horizonte de tempo para um data warehouse é significativamente maior do que o de sistemas operacionais.
  - Base de dados operacional: informação actual.
  - Dados no data warehouse: fornece informação numa perspectiva histórica (e.g., últimos 5-10 anos)
- Cada estrutura chave no data warehouse
  - Contém um elemento de tempo, explicita ou implicitamente
  - Mas a chave de dados operacionais pode ou não conter um "elemento de tempo".

Data Warehouse—Não-volátil

- Um repositório fisicamente separado de dados transformados do ambiente operacional.
- não ocorre actualização de dados operacional sobre a informação no data warehouse.
  - não requer mecanismos de processamento de transacções, recuperação e controlo de concorrência
  - Requer apenas duas operações de acesso a dados:
    - *Carregamento inicial de dados* e *acesso a dados*.

Data Warehouse vs. SGBD Heterogéneos

- Integração tradicional de BD heterogéneas:
  - Construir conversores/mediadores sobre BD heterogéneas
  - Abordagem orientada-a-consulta
    - Quando uma consulta é feita a uma determinada BD, usa-se um meta-dicionário para traduzir a consulta em consultas apropriadas para outras BD's envolvidas, e os resultados são integrados num conjunto resposta global
    - Filtragem de informação complexa, competição por recursos
- Data warehouse: orientada-por-actualizaçao, alta performance
  - A informação de fontes heterogéneas é previamente integrada e guardada em warehouses para consulta e análise directa

Data Warehouse vs. SGBD Operacionais

- OLTP (on-line transaction processing)
  - Tarefa principal dos SGBD relacionais tradicionais
  - Operaçoes diárias: vendas, inventário, saldos, produçao, salários, registo, contabilidade, etc.
- OLAP (on-line analytical processing)
  - Tarefa principal de sistemas de data warehouse
  - Análise de dados e tomada de decisões
- Características distintas (OLTP vs. OLAP):
  - Orientaçao do sistema e utilizador: cliente vs. mercado
  - Conteúdo dos dados: actuais, detalhados vs. históricos, consolidados
  - Desenho da BD: ER + aplicaçao vs. estrela + tema
  - Visao: actual, local vs. evolucionária, integrada
  - Padroes de acesso: actualização vs. consultas read-only, complexas

OLTP vs. OLAP

	OLTP	OLAP
utilizadores	Escriturário, profissional IT	Analista de mercado
funçao	Operaçoes diárias	Suporte a decisoes
desenho de BD	Orientado-por-aplicaçao	Orientado-por-tema
dados	correntes, actualizados detalhados, relacional simples isolado	históricos, sumarizados, multidimencionais integrados, consolidados
uso	repetitivo	ad-hoc
acesso	read/write index/hash na chave prim.	Leitura exaustiva
unid. de trabalho	Transacçao simples e curta	Consulta complexa
# registos acedidos	dezenas	Milhoes
#utilizadores	milhares	Centenas
tamanho da BD	100MB-GB	100GB-TB
métrica	Transacçoes por minuto	Consultas por minuto, resposta

Porquê Separar um Data Warehouse?

- Alta performance para ambos os sistemas
  - SGBD— optimizados para OLTP: métodos de acesso, indexação, controlo de concorrência, recuperação
  - Warehouse— optimizado para OLAP: consultas OLAP complexas, visoes multi-dimensionais, consolidação.
- Funções diferentes e dados diferentes:
  - Falta de dados: suporte à decisão requer dados históricos que BD's operacionais tipicamente não mantém
  - Consolidaçao de dados: SD requer consolidação (agregação, sumarizaçao) de dados de fontes heterogéneas
  - Qualidade de dados: Fontes diferentes usam tipicamente representações inconsistentes de dados, códigos e formatos que têm de ser reconciliados

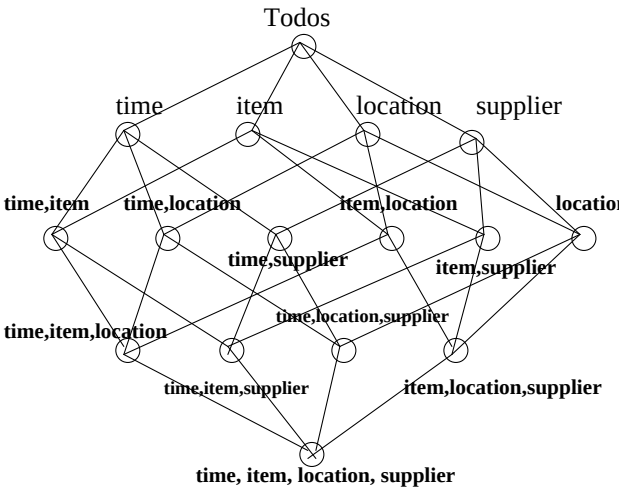
Data Warehousing e Tecnologia OLAP para Data Mining

- O que é um data warehouse?
- O modelo de dados multi-dimensional
- Arquitectura de data warehouses
- Implementação de data warehouses
- Mais aspectos da tecnologia multi-dimensional
- De data warehousing a data mining

De Tabelas e Folhas de Cálculo para Cubos de Dados

- Um data warehouse é baseado num modelo de dados multidimensional que vê os dados na forma de um cubo de dados
- Um cubo de dados, tal como sales, permite que a informação seja modelada e vista em múltiplas dimensões
  - Tabelas de dimensão, tais como item (item_name, brand, type), ou time(day, week, month, quarter, year)
  - Tabelas de factos contém medidas (tais como dollars_sold) e chaves externas para cada tabela de dimensão relacionada
- Na literatura de data warehousing, um cubo n-D é chamado cubóide. O cubóide 0-D de topo, que contém o nível mais alto de sumariazação, é chamado cubóide apex. O reticulado de cubóides forma o cubo de dados.

Cubo: Reticulado de cubóides



0-D(apex) cubóide

1-D cubóides

2-D cubóides

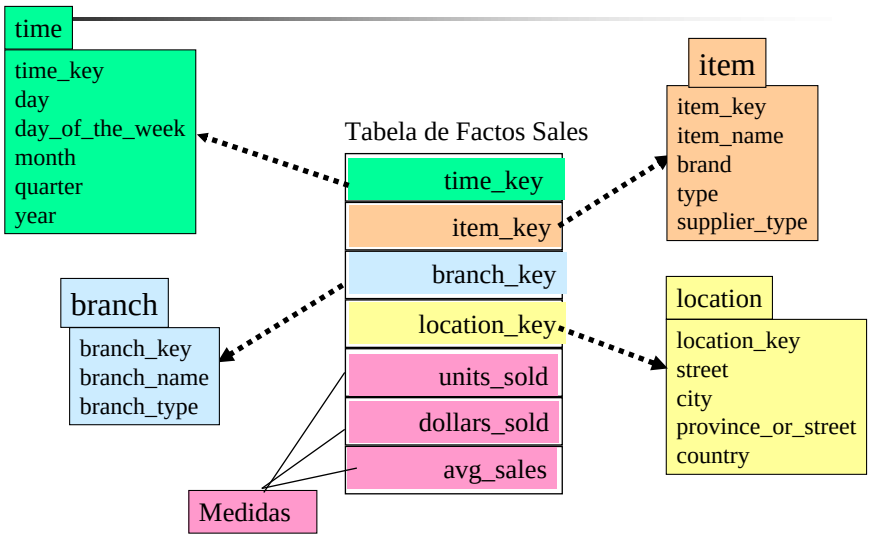
3-D cubóides

4-D(base) cubóide

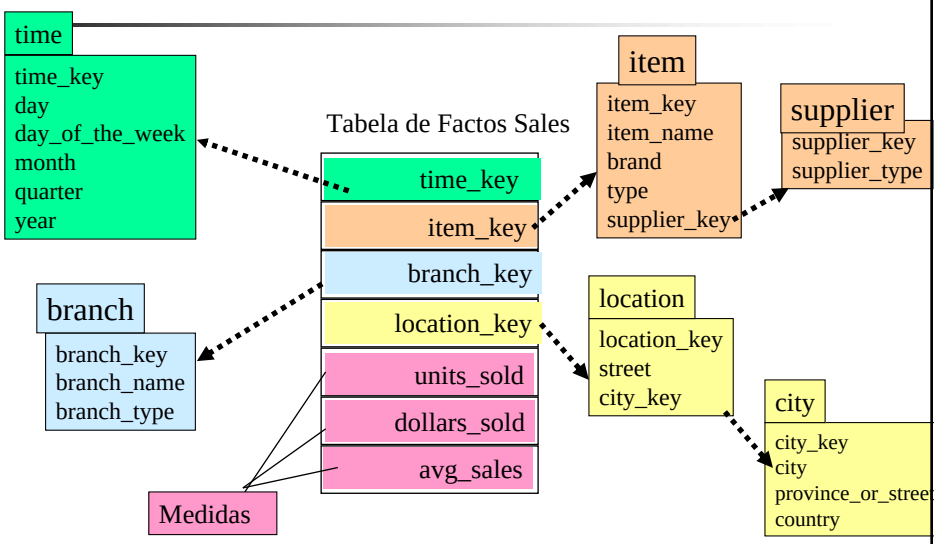
Modelação Conceptual de Data Warehouses

- Modelar data warehouses: dimensões & medidas
  - Esquema estrela: Tabela de factos no centro ligada a um conjunto de tabelas dimensão
  - Esquema floco de neve: Um refinamento do esquema estrela onde parte da hierarquia dimensional é normalizada num conjunto de tabelas dimensão mais pequenas, numa forma similar a um floco de neve.
  - Constelações de factos: Tabelas de factos múltiplas partilham tabelas dimensão, formando um grupo de estrelas, logo chamado constelação de factos.

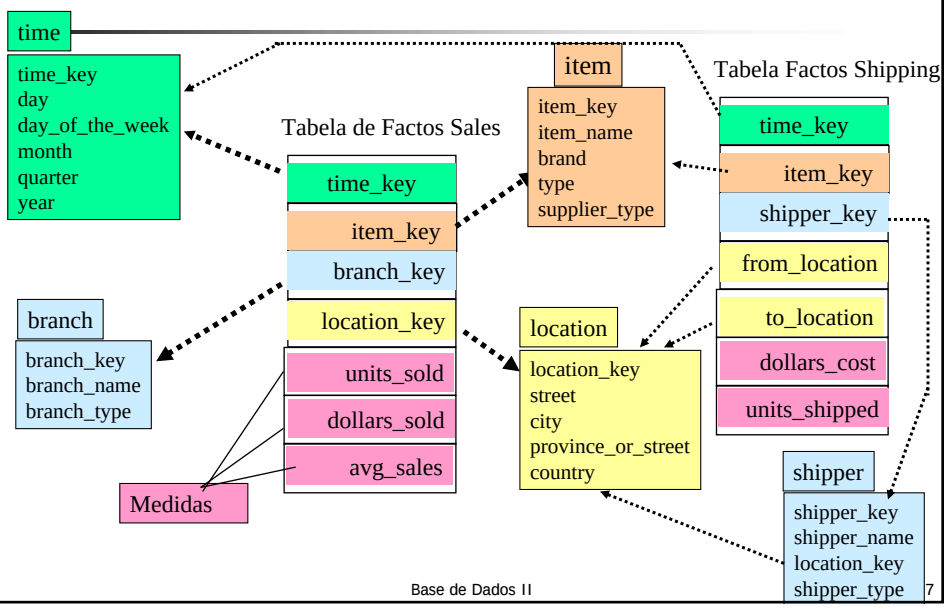
Exemplo de Esquema Estrela



Exemplo de Esquema Floco de Neve



Exemplo de Constelação de Factos



Data Mining Query Language, DMQL: Primitivas da Linguagem

- Definição de Cubo (Tabela de Factos)

  define cube <cube_name> [<dimension_list>]: <measure_list>
- Definição de Dimensão (Tabela de Dimensao)

  define dimension <dimension_name> as (<attribute_or_subdimension_list>)
- Caso Especial (Tabelas de dimensão partilhadas)
  - Primeira vez como “definição de cubo”
  - define dimension <dimension_name> as <dimension_name_first_time> in cube <cube_name_first_time>

Definição de Esquema Estrela em DMQL

define cube sales_star [time, item, branch, location]:
dollars_sold = sum(sales_in_dollars), avg_sales = avg(sales_in_dollars), units_sold = count(*)
define dimension time as (time_key, day, day_of_week, month, quarter, year)
define dimension item as (item_key, item_name, brand, type, supplier_type)
define dimension branch as (branch_key, branch_name, branch_type)
define dimension location as (location_key, street, city, province_or_state, country)

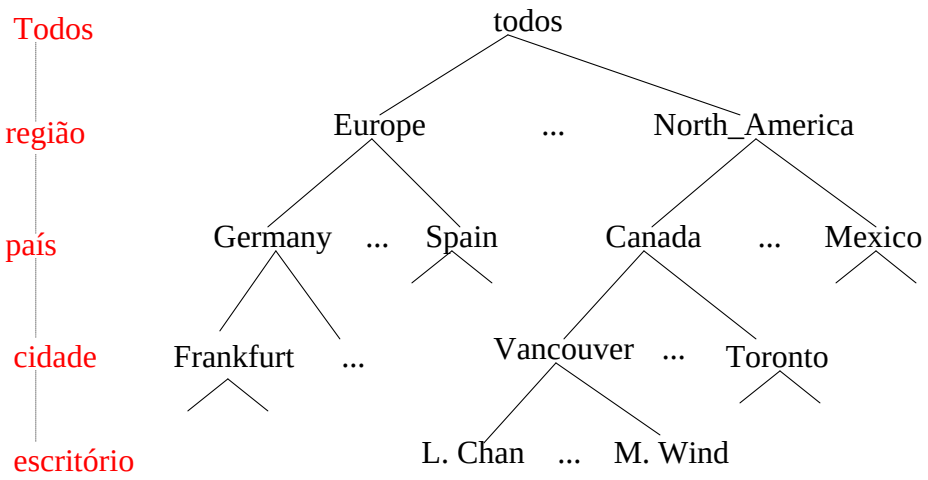
Definição de esquema Floco de Neve em DMQL

define cube sales_snowflake [time, item, branch, location]:
dollars_sold = sum(sales_in_dollars), avg_sales = avg(sales_in_dollars), units_sold = count(*)
define dimension time as (time_key, day, day_of_week, month, quarter, year)
define dimension item as (item_key, item_name, brand, type, supplier(supplier_key, supplier_type))
define dimension branch as (branch_key, branch_name, branch_type)
define dimension location as (location_key, street, city(city_key, province_or_state, country))

Definição de constelação de factos em DMQL

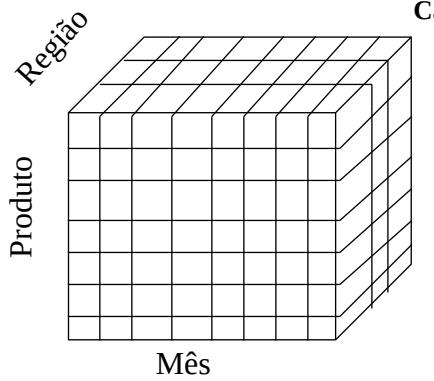
```
define cube sales [time, item, branch, location]:
        dollars_sold = sum(sales_in_dollars), avg_sales =
          avg(sales_in_dollars), units_sold = count(*)
define dimension time as (time_key, day, day_of_week, month, quarter, year)
define dimension item as (item_key, item_name, brand, type, supplier_type)
define dimension branch as (branch_key, branch_name, branch_type)
define dimension location as (location_key, street, city, province_or_state,
    country)
define cube shipping [time, item, shipper, from_location, to_location]:
        dollar_cost = sum(cost_in_dollars), unit_shipped = count(*)
define dimension time as time in cube sales
define dimension item as item in cube sales
define dimension shipper as (shipper_key, shipper_name, location as location
    in cube sales, shipper_type)
define dimension from_location as location in cube sales
define dimension to_location as location in cube sales
```


Hierarquias conceptuais: Dimensão (localização)

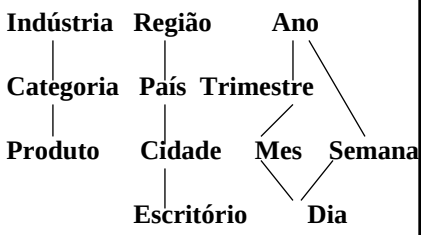


Dados Multi-dimensionais

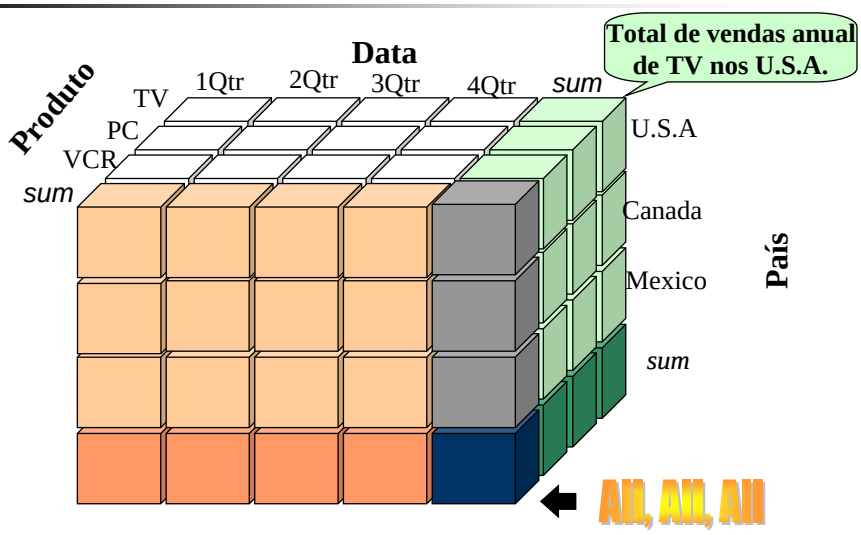
- Volume de vendas como função de produto, mês, região



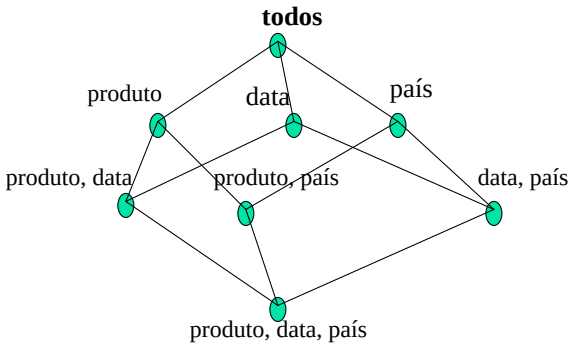
Dimensões: Produto, Localização, Tempo
Caminhos de sumarização hierárquicos



Exemplo de Cubo de Dados



Cubóides correspondentes ao Cubo



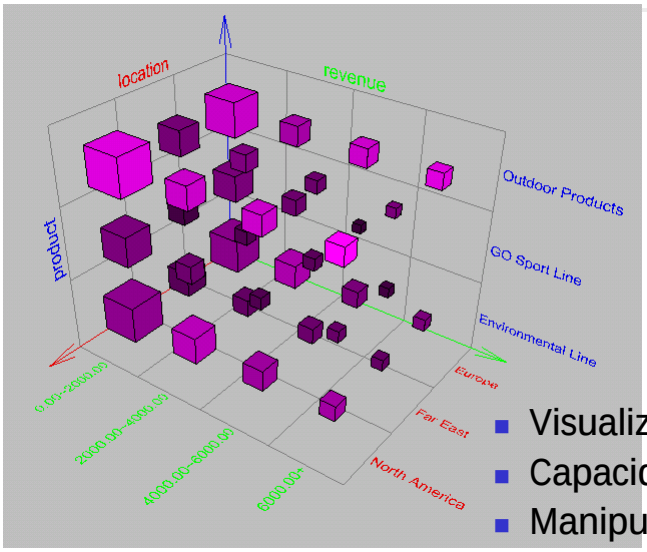
0-D(apex) cubóide

1-D cubóides

2-D cubóides

3-D(base) cubóide

Pesquisa num Cubo de Dados



- Visualização
- Capacidades OLAP
- Manipulação interactiva

Operações OLAP típicas

- Roll up (drill-up): sumarizar dados
  - *por subida na hierarquia ou por redução de uma dimensão*
- Drill down (roll down): inverso de roll-up
  - *de sumários de nível mais alto para sumários de nível mais baixo ou mais detalhados, ou pela introdução de dimensões*
- Slice and dice:
  - *project e select*
- Pivot (rotate):
  - *reorientar o cubo, visualização, de 3D para séries de planos 2D*
- Outras operações
  - *drill across: envolvem mais do que uma tabela de factos*
  - *drill through: do nível mais baixo do cubo para as tabelas relacionais de back-end (usando SQL)*

Data Warehousing e Tecnologia OLAP para Data Mining

- O que é um data warehouse?
- O modelo de dados multi-dimensional
- Arquitectura de data warehouses
- Implementação de data warehouses
- Mais aspectos da tecnologia multi-dimensional
- De data warehousing a data mining

Desenho de Data Warehouses

- Quatro perspectivas de desenho de um data warehouse
  - Perspectiva Top-down
    - Permite a selecção da informação relevante necessária para o data warehouse
  - Perspectiva de Origem de Dados
    - Mostra a informação a ser adquirida, guardada e gerida por sistema operacionais
  - Perspectiva Data warehouse
    - consiste em tabelas de factos e tabelas dimensão
  - Perspectiva de Consulta de Análise
    - vê a perspectiva dos dados no warehouse do ponto de vista do utilizador final

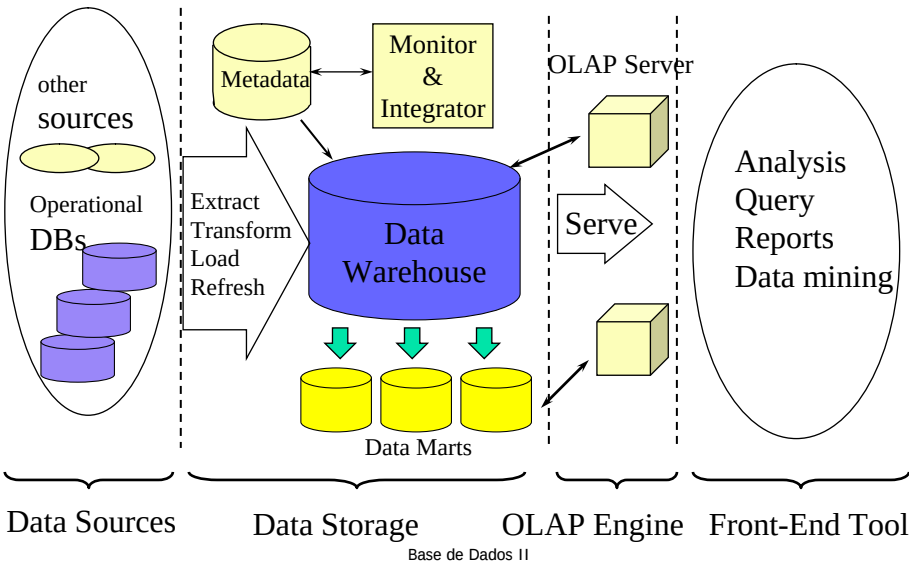
Processo de desenho de Data Warehouses

- Abordagens Top-down, bottom-up ou uma combinaçao de ambos
  - Top-down: Começa com o desenho e planeamento geral
  - Bottom-up: Começa com experencias e prototipos
- Do ponto de vista da engenharia de software
  - Cascata: Análise estruturada e sistematica em cada passo antes de prosseguir para o proximo
  - Espiral: Geraçao rapida e incremental de funcionalidades do sistema
- Processo de desenho típico de data warehouse
  - Escolher um processo de negócio a modelar, e.g., encomendas, facturas, etc.
  - Escolher o *grao* (nível de dados atómico) do processo de negócio
  - Escolher as dimensoes que estao associadas a cada tabela de factos
  - Escolher as medidas presentes em cada registo da tabela de factos

Modelação de um Data Warehouse

- Suponha que uma empresa concessionária de auto-estradas pretende construir uma BD para suportar o registo das viagem efectuadas.
  - Crie um Modelo relacional para uma BD operacional
  - Crie um Modelo estrela para um DW

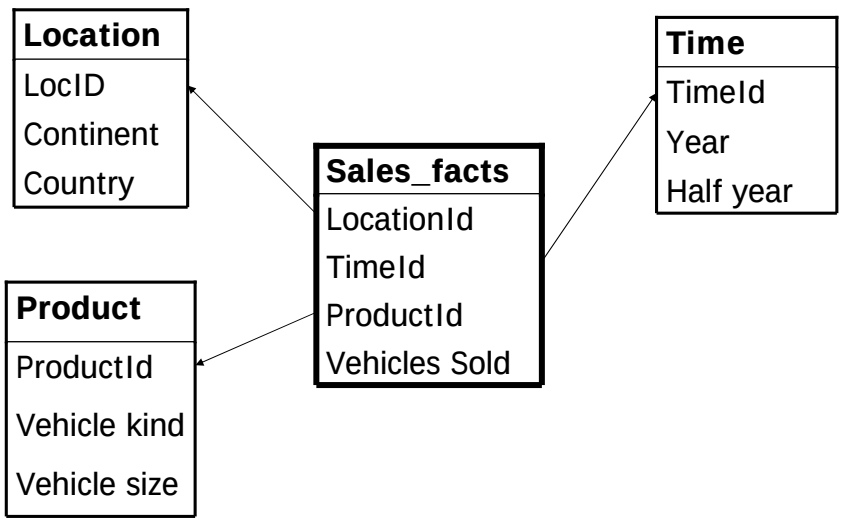
Arquitectura Multi-Camada



Arquitecturas OLAP

- Relational OLAP (ROLAP)
  - Usar SGBD relationais ou relacionais extendidos para guardar e gerir os dados do data warehouse e aplicaçoes OLAP para suportar análise complexa de dados.
  - Incluem optimizaçao dos SGBD de backend, implementaçao de navegaçao baseada em agregaçao, assim como mais ferramentas e serviços.
  - Maior escalabilidade
- Multidimensional OLAP (MOLAP)
  - Baseado em matrizes de armazenamento multidimensionais (sparse matrix techniques)
  - Indexaçao rápida sobre dados sumarizados pre-cálculados
- Hybrid OLAP (HOLAP)
  - Baixo nível: relacional, alto-nível: matriz

Esquema Estrela para Vendas de Veículos



Representação relacional do cubo

Location Dimension Table

Location_Key	Continent	Country
1	Europe	Germany
2	Europe	Spain
3	America	USA
4	America	Canada

Time Dimension Table

Time_Key	Year	Half_Year
1	1996	1HF
2	1996	2HF
3	1997	1HF
4	1997	2HF

Product Dimension Table

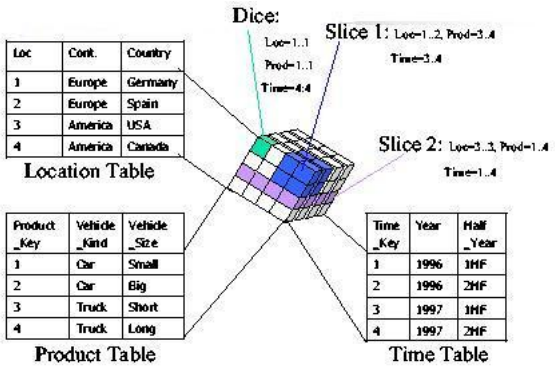
Product_Key	Vehicle_Kind	Vehicle_Size
1	Car	Small
2	Car	Big
3	Truck	Short
4	Truck	Long

Representação relacional do cubo

Fact Table

Location_Key	Product_Key	Time_Key	Vehicles_Sold
1(Eur,Ger)	1(Car,Small)	1(1996,1HF)	2450
1(Eur,Ger)	1(Car, Small)	2(1996,2HF)	2890
1(Eur,Ger)	1(Car, Small)	3(1997,1HF)	2650
1(Eur,Ger)	1(Car, Small)	4(1997,2HF)	800
...	...	...	...
...	...	...	...
4(Ame,Can)	4(Truck,Long)	4(1997,2HF)	500

O cubo OLAP



Loc	Cont.	Country
1	Europe	Germany
2	Europe	Spain
3	America	USA
4	America	Canada

Location Table

Product _Key	Vehicle _Kind	Vehicle _Size
1	Car	Small
2	Car	Big
3	Truck	Short
4	Truck	Long

Product Table

Time _Key	Year	Half _Year
1	1996	1HF
2	1996	2HF
3	1997	1HF
4	1997	2HF

Time Table

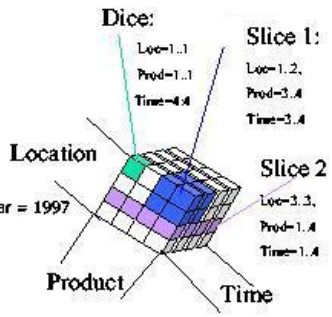
Expressões SQL para as operações OLAP

- Slice 1

```
SELECT Vehicles_Sold
FROM Location L, Product P, Time T, Fact F
WHERE L.Location_Key = F.Location_Key
   AND P.Product_Key = F.Product_Key
   AND T.Time_Key = F.Time_Key
   AND  L.Continent = 'Europe'
   AND P.Vehicle_Kind = 'Truck' AND  T.Year = 1997
```

- Slice 2

```
SELECT Vehicles_Sold
FROM Location L, Product P, Time T, Fact F
WHERE L.Location_Key = F.Location_Key
   AND P.Product_Key = F.Product_Key
   AND T.Time_Key = F.Time_Key
   AND L.Country = 'USA'
```



Roll Up and Drill Down can be implemented by Group By and aggregate functions

Expressões SQL para as operações OLAP

- *"Analyzing the sales by continents"*

```
SELECT SUM(F.Vehicles_Sold)
FROM Location L, Product P, Time T, Fact F
WHERE L.Location_Key = F.Location_Key
   AND P.Product_Key = F.Product_Key
   AND T.Time_Key = F.Time_Key
GROUP BY L.Continent
```

- *"Analyzing the sales by countries"*

```
SELECT SUM({F.Vehicles_Sold)
FROM Location L, Product P, Time T, Fact F
WHERE L.Location_Key = F.Location_Key
   AND P.Product_Key = F.Product_Key
   AND T.Time_Key = F.Time_Key
GROUP BY L.Continent, L.Country0
```

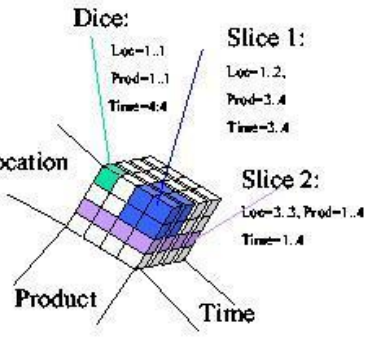

Representação matricial do cubo

- MOLAP uses matrix or array to represent Fact tables as in ROLAP
  - Fact(Loc, Prod, Time)
- Dice
  - Fact(1,1,4)
- Slice 1

$$I^2_{Loc=1} I^4_{Prod=3} K^4_{Time=3} Fact(Loc, Prod, Time)$$

- Slice 2

$$I^3_{Loc=3} I^4_{Prod=1} K^4_{Time=1} Fact(Loc, Prod, Time)$$



Operador Cube

- Definiçao do cubo e cálculo em DMQL

  define cube sales[item, city, year]: sum(sales_in_dollars)

  compute cube sales

- Transformar numa linguagem tipo SQL- (com um novo operador cube by, introduzido por Gray et al.'96)

  SELECT item, city, year, SUM (amount)

  FROM SALES

  CUBE BY item, city, year

- E necessario calcular os seguintes Group-Bys

  (date, product, customer),
  (date,product),(date, customer), (product, customer),
  (date), (product), (customer)
  ()