



Sistemas de informação

enterprise analytics and data
warehousing

Profº Fabiano J. Cury Marques

enterprise analytics and data warehousing

Modelagem Dimensional Parte IV



Agenda

- ✕ Normalização de dimensões – Snowflaking
- ✕ Erro de Projeto: Muitas dimensões
- ✕ Referências

1.

SnowFlaking

Normalização de dimensões – Snowflaking



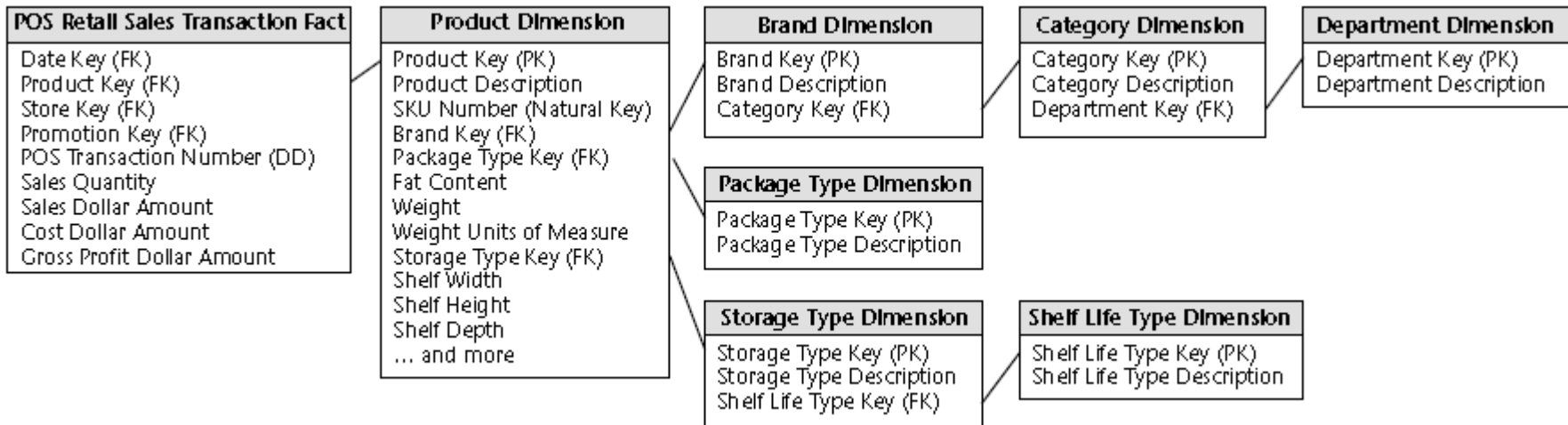
Snowflaking

- ✗ As tabelas dimensão desnormalizadas com repetições de valores textuais podem fazer com que os projetistas de base de dados sintam-se desconfortáveis
- ✗ Ex.: **150.000 produtos** pertencem a apenas **50 departamentos** distintos. Em vez de armazenar redundantemente a **descrição do departamento com 20 bytes** na dimensão Produto, projetistas com bagagem de normalização querem armazenar um **código de departamento com 2 bytes** apenas e então criar uma nova tabela dimensão Departamento que irá decodificar os departamentos
- ✗ De fato, para estes projetistas, isso serviria para todos os descritores de nossas tabelas
- ✗ Além disso tem o fator da manutenção, mas lembre-se, no DW, isso ocorre lá no repositório Stage e não no DDS



Snowflaking

- ✗ A normalização das tabelas dimensão tipicamente são chamadas de **snowflaking**
- ✗ Atributos redundantes são removidos da tabela dimensão desnormalizada e colocados em tabelas dimensão secundárias





Snowflaking

- ✗ Apesar do snowflaking ser uma extensão legal do modelo dimensional, em geral, encoraja-se a resistir a tentação de normalizar seu modelo dimensional dado nossas duas principais premissas de projeto:
 - Facilidade de Uso
 - Desempenho

- ✗ As tabelas snowflaking tornam a representação muito mais complexa. Além disso, os otimizadores das bases de dados terão muito mais trabalho com inúmeros joins, gerando maior lentidão no desempenho das consultas

- ✗ Se o problema for espaço observe que a **economia de uma dimensão snowflaked é mínimo**: no nosso exemplo anterior, $150.000 * 20$ bytes trocados por $150.000 * 2$ bytes economizam APENAS 2,7 MB. Enquanto isso temos fatos ocupando facilmente 10 GB

Snowflaking



✗ Dilema de modelagem ...

VOLUME DE DADOS
ARMAZENADO!

VS

DESEMPENHO NA
CONSULTA DOS
DADOS

FACILIDADE DE USO

- ✗ Se o problema for espaço observe que a **economia de uma dimensão snowflaked é mínimo**: no nosso exemplo anterior, $150.000 * 20$ bytes trocados por $150.000 * 2$ bytes economizam APENAS 2,7 MB. Enquanto isso temos fatos ocupando facilmente 10 GB



3.

Muitas Dimensões

Muitas Dimensões



Muitas Dimensões

- ✗ A tabela fato no esquema dimensional é naturalmente altamente normalizada e compacta
- ✗ Não há como normalizar mais ainda relacionamentos **muitos-para-muitos** complexos pois as dimensões não são correlacionadas
- ✗ É interessante notar que, apesar de desconfortáveis com tabelas dimensão desnormalizadas, alguns projetistas são tentados a desnormalizar a tabela fato
- ✗ Em vez de ter a chave estrangeira de um produto específico, inclui-se chaves estrangeiras para elementos analisados com frequência na hierarquia de produtos. Ex.: marca, categoria e departamento



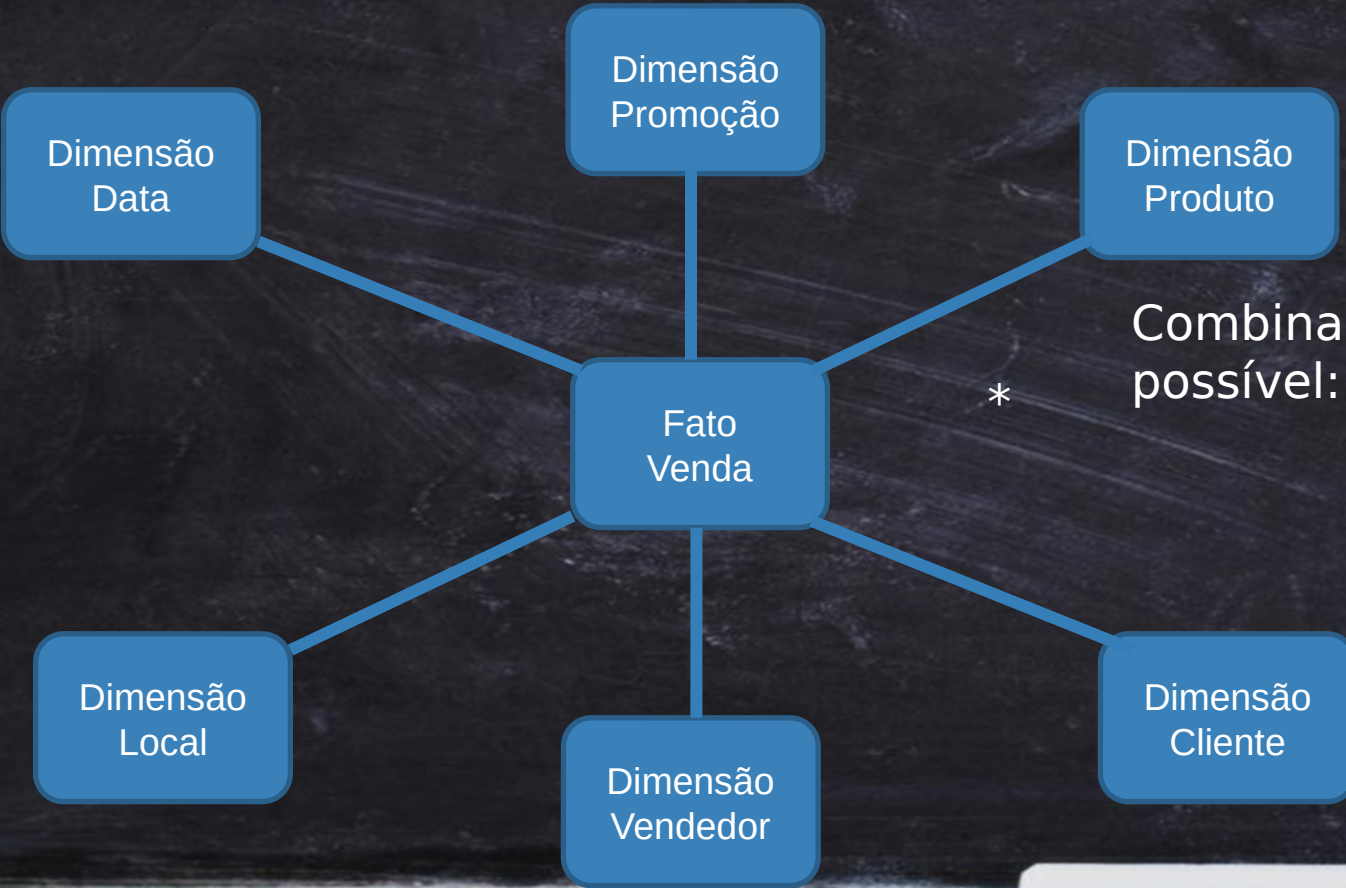
Muitas Dimensões

- ✘ Antes que perceba, sua tabela fato que deveria ser compacta já está relacionando dezenas de tabela dimensão
- ✘ Normalmente as tabelas com estas características são chamadas de **centopéia**
- ✘ Lembre-se que apesar de não estarmos preocupados com o **espaço consumido pelas tabelas dimensão**, as tabelas fato sim devem ter esse fator levado em consideração
- ✘ A maior parte dos processos de negócio podem ser representados com **menos de 15 dimensões na tabela fato**. Se passar muito disso, deve-se verificar se não há como combinar **dimensões correlacionadas** em uma só (ex.: níveis hierárquicos)



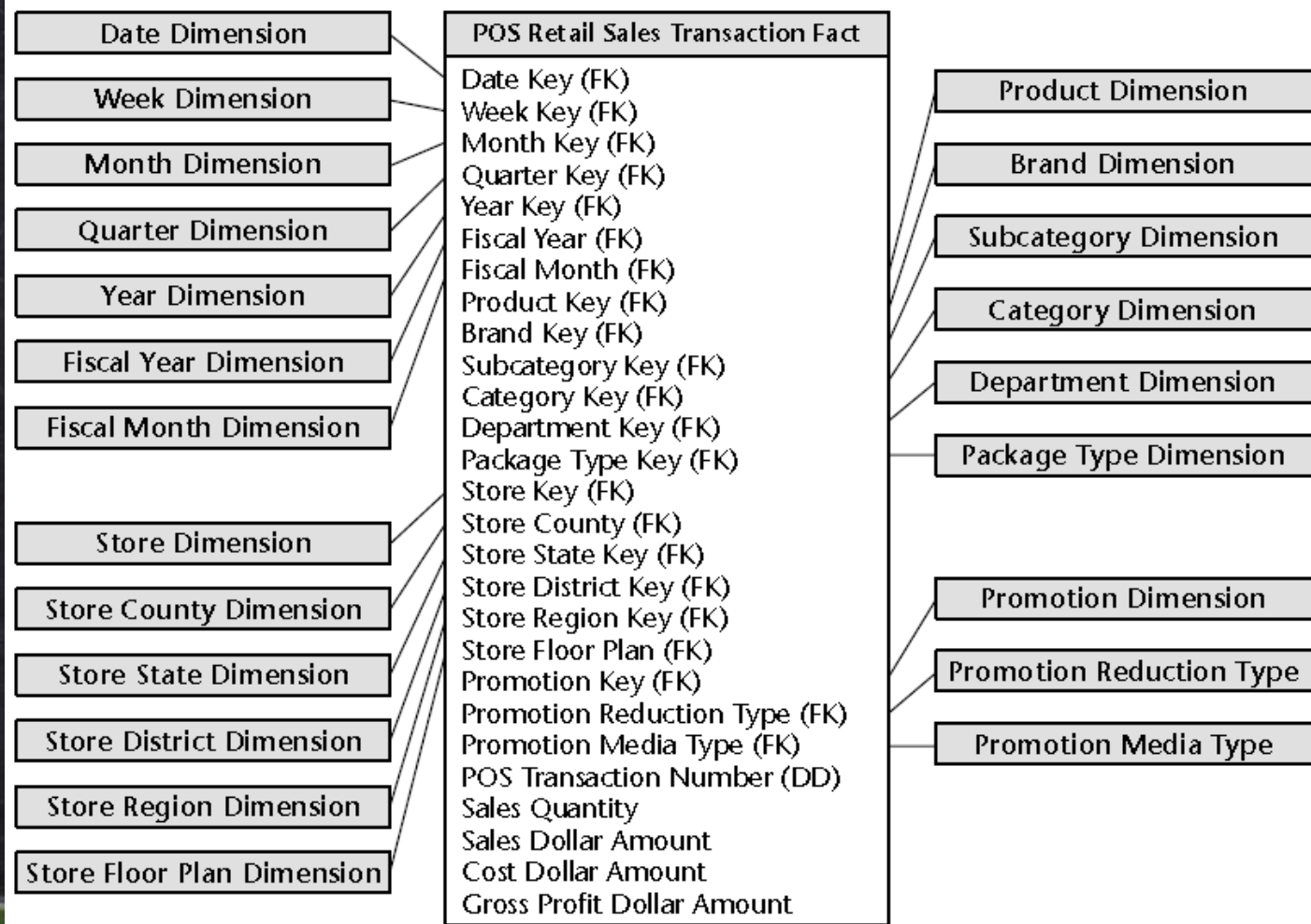


Muitas Dimensões



*
Combinação máxima possível:

Muitas Dimensões



QUANTIDADE DE DIMENSÕES



✕ Dilema de modelagem ...

EXPRESSIVA
QUANTIDADE DE
CHAVES
EXTRANGEIRAS NA
TABELA FATO

VS

DIVERSIDADE DE
DIMENSÕES

2.

scd

Slowly Changing Dimension

SLOWLY CHANGING DIMENSION (SCD)



- ✗ SCD é uma técnica utilizada para armazenar os valores históricos de atributos de uma dimensão
- ✗ Os valores dos atributos dimensionais mudam com o tempo. Quando eles mudam, você pode sobrescrever os valores antigos com os novos, ou pode preservar o valor antigo
- ✗ Existem dois métodos para se preservar o valor antigo:
 - ✗ Armazená-los como linhas
 - ✗ Armazená-los como colunas



SLOWLY CHANGING DIMENSION (SCD)

- ✗ Suponha que a loja 7 estava na região 1, mas agora está na região 2. Você quer armazenar a informação histórica, o fato de que ela foi da região 1
- ✗ Você pode fazer isso armazenando a informação como linhas ou coluna
- ✗ Como linha:

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

- ✗ Como coluna:

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

- Neste caso a questão é, o que fazer se a região mudar novamente?
- Você pode criar mais uma coluna old_region2
- Esta é a principal desvantagem em relação ao anterior



TIPOS DE SCD

- ✖ SCD tipo 1
 - Sobreescreve os valores antigos dos atributos, assim, os valores antigos não são mantidos

- ✖ SCD tipo 2
 - Mantém os valores antigos criando uma nova linha para cada mudança

- ✖ SCD tipo 3
 - Mantém os valores antigos os colocando em outra coluna

SLOWLY CHANGING DIMENSION (SCD)



✕ Dilema de modelagem ...

**ARMAZENAMENTO
DE DADOS POR
LINHA OU
COLOUNA**

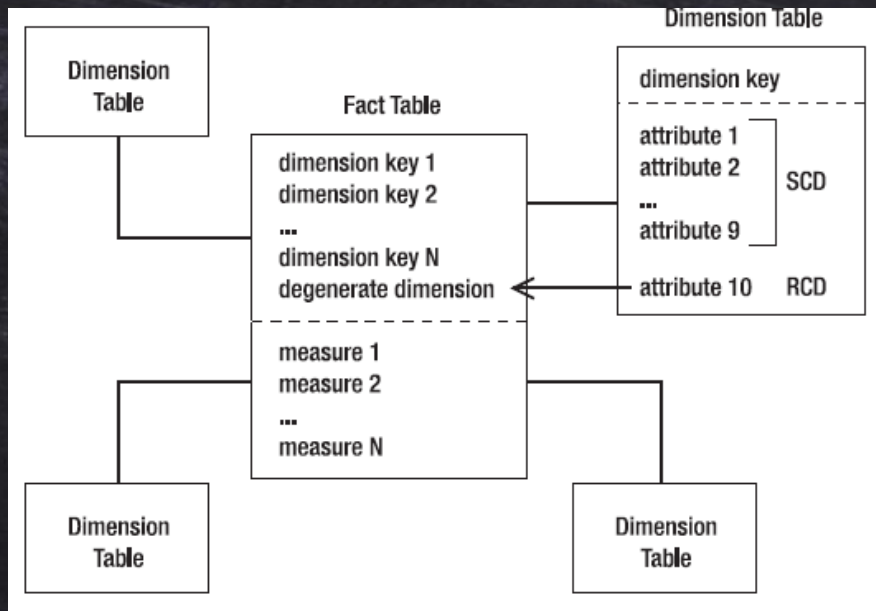
VS

FLEXIBILIDADE DE
CONSTRUÇÃO DE
CONSULTAS PARA
ACESSO AO DADO

E A RAPIDLY CHANGING DIMENSIONS (RCD)?



- ✗ Um critério para definir se um atributo muda lentamente é o tempo de sua atualização, algo como uma vez a cada trimestre, bimestre no mínimo
- ✗ Mais do que isso, não é SCD, então trata-se diferentemente, tirando o atributo que muda com grande frequência da tabela dimensão e colocando diretamente na fato



RAPIDLY CHANGING DIMENSIONS (RCD)



✗ Dilema de modelagem ...

**ALTERAÇÃO RAPIDA
DE DADOS NA
TABELA DIMENSÃO**

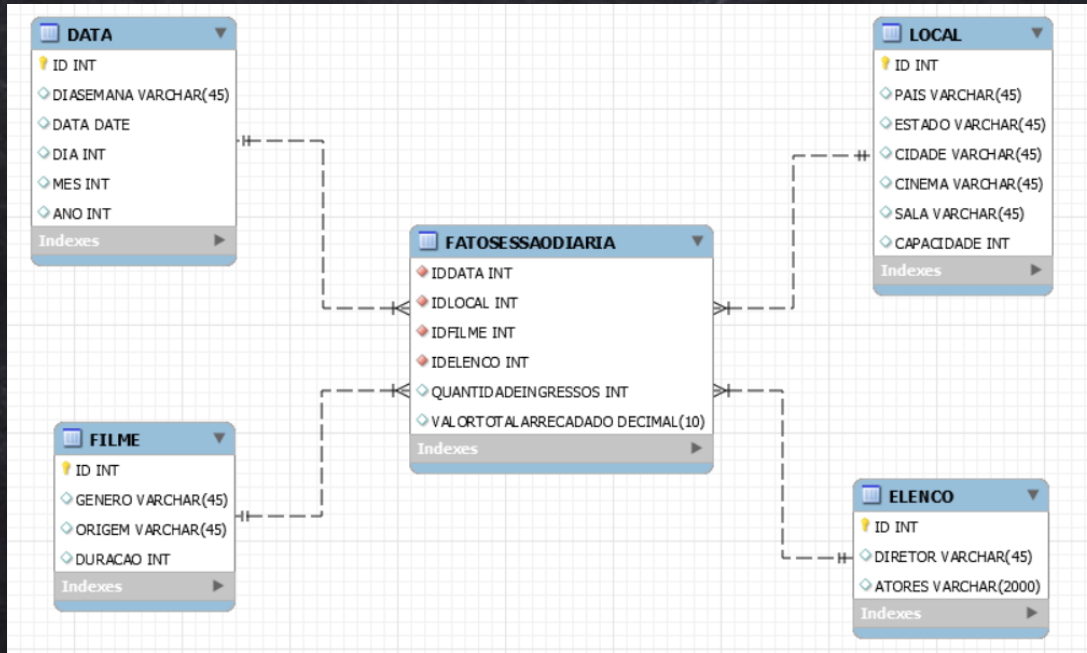
VS

DEFINIÇÃO DE
NOVA MEDIDA NA
TABELA FATO



Exercício:

- ✗ De forma livre, sugira um redesenho do DataMart abaixo para ao menos uma situação hipotética que possa utilizar snowflaking:





Referências

- ✕ KIMBALL, R., ROSS, M. The Data Warehouse Toolkit. 2ª ed., John Wiley Professional, 2002.
- ✕ MACHADO, F. N. R. Tecnologia e Projeto de Data Warehouse. 1ª ed., São Paulo: Ed. Érica, 2004.



Obrigado!

Copyright © 2019 Prof. MSc. Eng. Wakim B. Saba

<https://br.linkedin.com/in/wakimsaba>

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).