Esse documento tende a explicar e esclarecer dúvidas sobre o modelo aplicado ao Bancada Analítica.

GFT

Assuntos abordados:

- ✓ Kimball x Inmon
- ✓ Transacional x DW
- ✓ EDW ou ODS
- ✓ Melhores Práticas no Hadoop
- ✓ Outras abordagens
- ✓ Data Lake
- ✓ Metodologia aplicada ao Bancada

Kimball X Inmon

GFT

Ralph Kimball versus Bill Inmon Comparison

	Kimball	Inmon
Need	Immediate	Longer time scale
Drive	Business areas	Enterprise
Budget	Smaller budget	Larger budget
Requirements	Volatile	More stable and growing
Customer	User base	Corporate
Sources	Stable	Changeable
Startup cost	Lower	Higher
Projects	Same cost as start up	Cheaper than start up

Cada um acredita(va) estar certo e o outro errado!

	Inmon	Kimball
Source Required	✓	✓
Staging	✓	✓
ETL	✓	✓
Data Marts	✓	✓
Business Requirements	✓	✓
Time attribute of data	✓	✓
Enterprise DW	✓	X
Dimensional tools	X	✓
Relational tools	✓	X
Process oriented	X	✓
Normalized data model	✓	X
Complex to design	✓	X
Continuous and Discrete Time Frame	✓	X
Slowly changing Time Frame	X	✓

Veja que no modelo Kimball não existe a figura do EDW.

Pontos Positivos e Negativos

GFT

Ambos os modelos têm os seus pontos positivos e negativos.

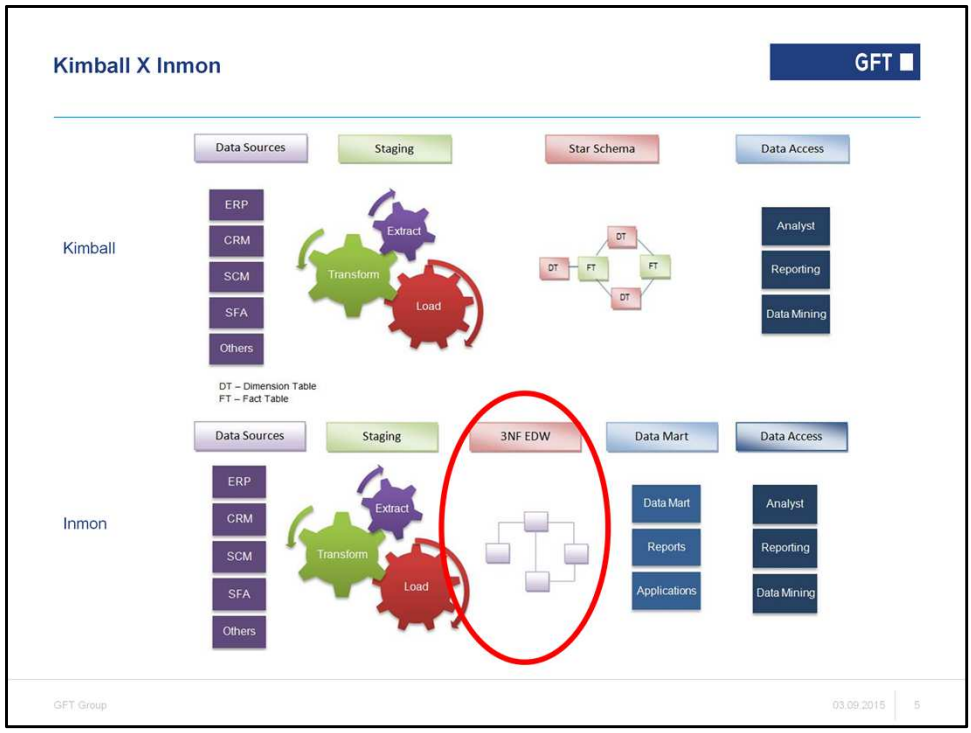
O modelo de Kimball Bottom-Up é mais escalável por causa da abordagem bottom-up e, portanto, você pode começar pequeno e ir crescendo progressivamente.
O retorno do investimento é mais rápido com o modelo de Kimball.

Por outro lado, o modelo de Inmon Top-Down é mais estruturado e mais fácil de manter, ao mesmo tempo em que é rígido e leva mais tempo para construir.
A vantagem significativa do modelo de Inmon é porque o DW é em 3NF; ficando mais fácil construir modelos de mineração de dados.

Ambos os modelos Kimball e Inmon concordam e enfatizam que DW é o repositório central de dados, e cubos OLAP são construídos de esquemas denormalizados estrela.

Em conclusão, é irrelevante a quem pertence, desde que você entenda porque está adotando um modelo específico.

No caso do Bancada Analítica faz mais sentido ter uma abordagem híbrida, entregando mais rápido, mas sempre pensando em integrar o todo.



No modelo Kimball não existe EDW.

O Datamart abrange uma determinada área de assunto, oferecendo informações mais detalhadas sobre o mercado (ou departamento) em questão.

Quando comparado um assunto no DW e no Datamart a diferença é a quantidade de informações a serem exibidas, uma vez que somente as informações de comitentes chegam a 250 colunas.

Kimball – Sem EDW

GFT

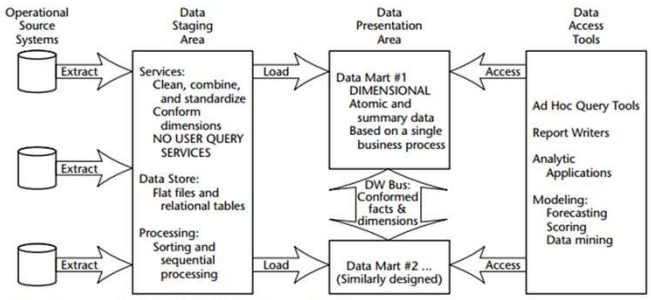


Figure 1.1 Basic elements of the data warehouse.

A única estrutura em 3 forma normal no modelo bottom-up é a própria stage.

No DW existem dimensões conformadas que podem ser compartilhadas entre os demais assuntos armazenados.

- Fonte: The Data Warehouse Toolkit Second Edition – Ralph Kimball

Transacional X DW

GFT

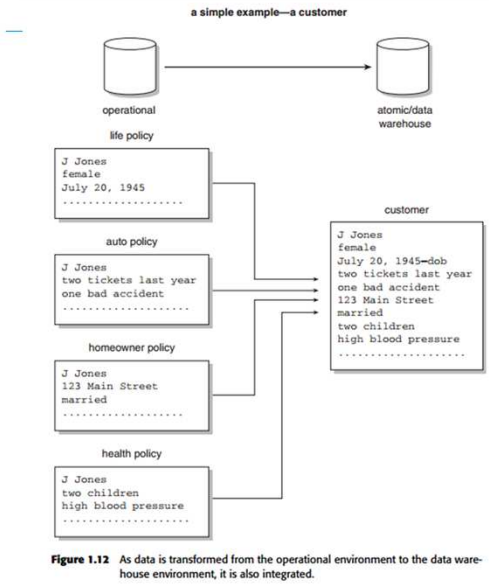


Figure 1.12 As data is transformed from the operational environment to the data warehouse environment, it is also integrated.

Nesse exemplo clássico, podemos ver que a dimensão cliente foi criada a partir da denormalização das 4 tabelas à esquerda, facilitando assim o acesso a informação e diminuindo o trabalho de joins.

- Fonte: Building the DataWarehouse Third Edition – Bill Inmon

EDW ou ODS?

GFT

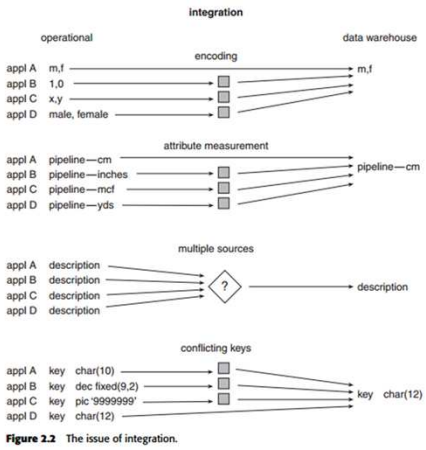


Figure 2.2 The issue of integration.

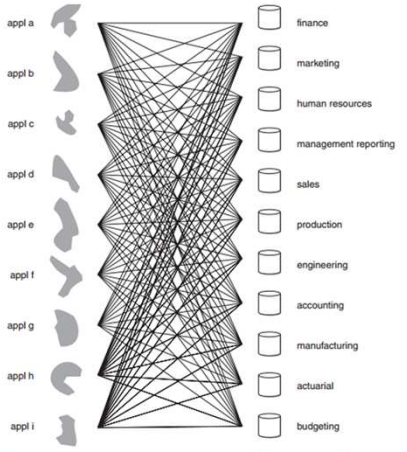
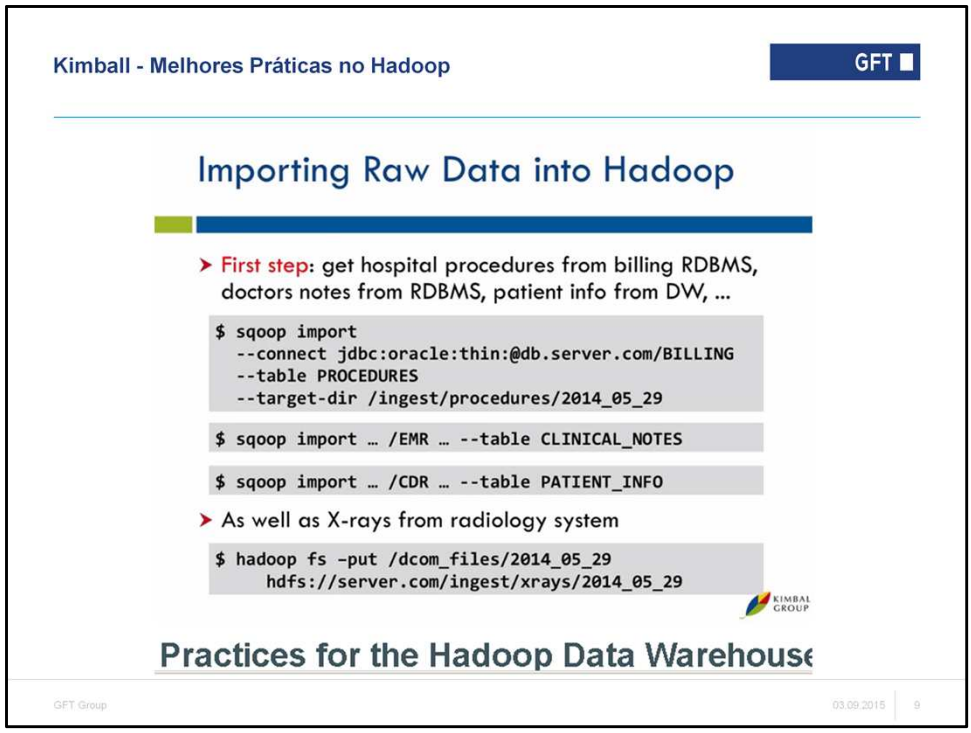


Figure 5.8 There are many applications, and there are many data marts. An interface application is needed between each occurrence. The result of bypassing the

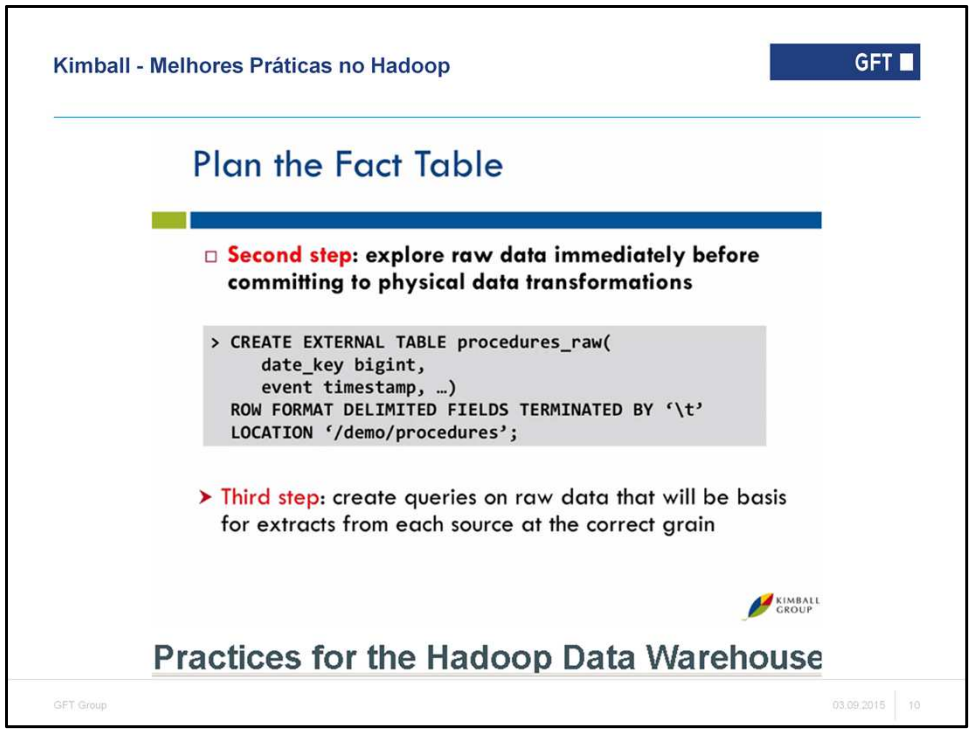
- Fonte: Building the DataWarehouse Third Edition – Bill Inmon

A utilização de um EDW se torna realmente necessária quando existe a necessidade de integrar diversas aplicações, como no exemplo (A,B,C,D), criando tabelas de "de-para" para unificar informações com o mesmo significado, ou mesmo integrar sistemas como atendimento ao cliente e vendas, por exemplo.
A CETIP possui um único banco de dados Oracle com todas as aplicações.



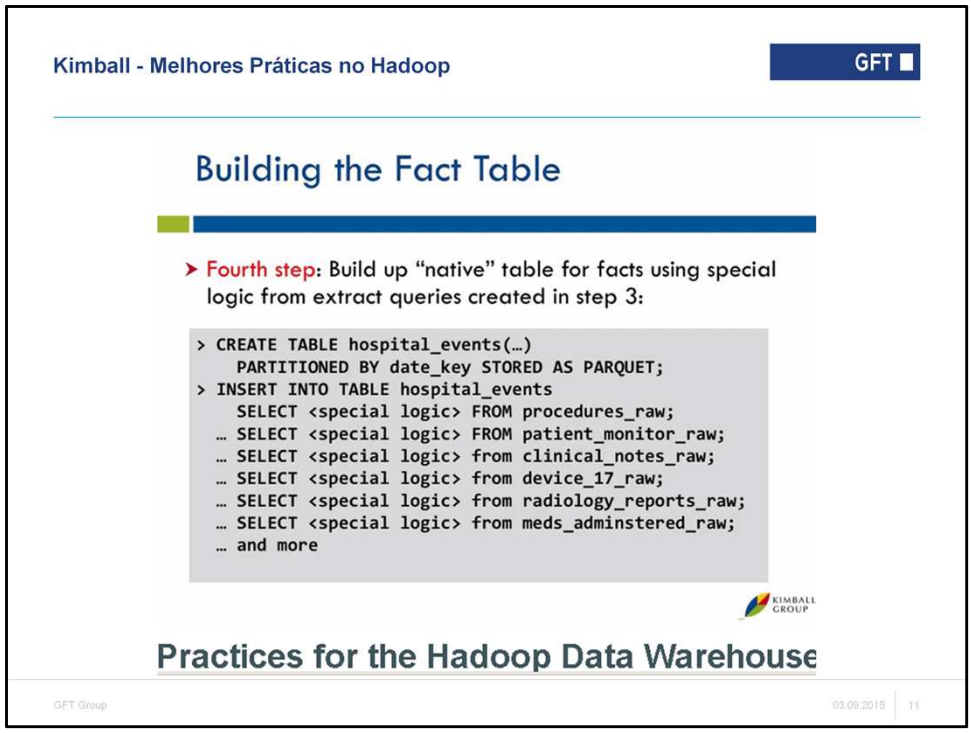
Nesse exemplo, podemos notar que a tabela PROCEDURES é carregada em arquivos no diretório procedures, com partições por data. Ela mantém a mesma estrutura da origem, contudo com n dias de histórico, isso já na stage.

No caso do Bancada Analítica, não temos como filtrar informações na origem por data de inclusão e/ou data de alteração, devido a manipulação onde nem sempre essas datas são atualizadas.
Gerar cópias da stage todos os dias, só duplicaria registros diariamente em partições diferentes.
As inclusões e alterações são armazenadas sempre no DWH, denormalizando os dados e garantindo que só existam novas fotos do que foi incluído e/ou alterado.



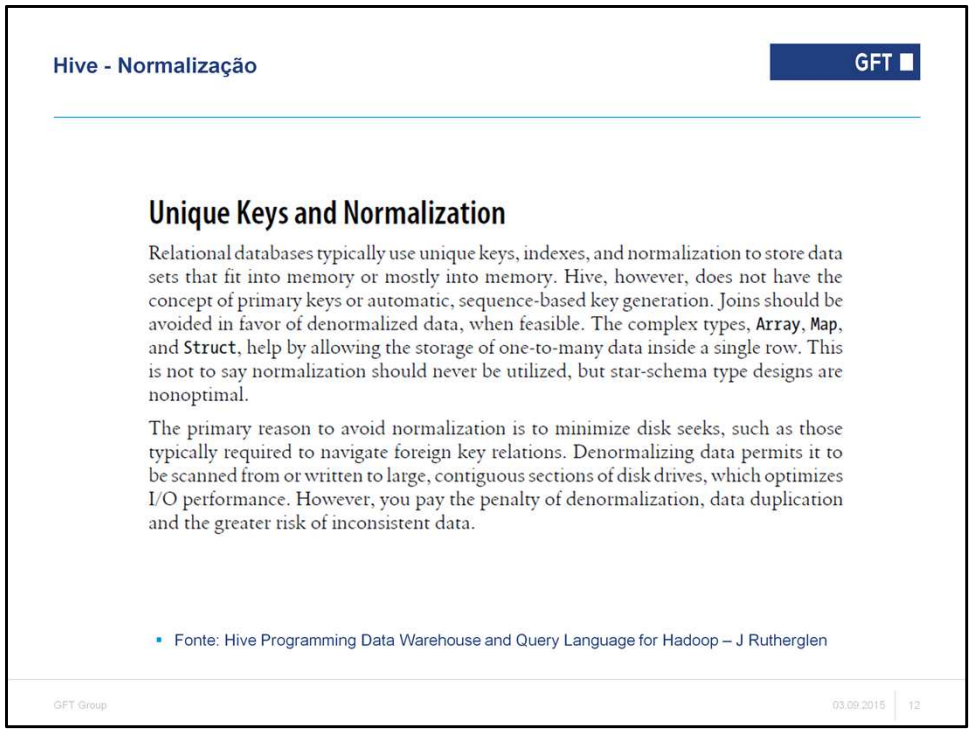
Aqui a tabela PROCEDURES_RAW é criada apontando para o diretório onde os arquivos foram salvos no slide anterior (stage).
Nesse momento, temos uma tabela com o histórico de dados de procedimentos.

No caso do Bancada Analítica, não temos como filtrar informações na origem por data de inclusão e/ou data de alteração, devido a manipulação onde nem sempre essas datas são atualizadas.
Gerar cópias da stage todos os dias, só duplicaria registros diariamente em partições diferentes.
As inclusões e alterações são armazenadas sempre no DWH, denormalizando os dados e garantindo que só existam novas fotos do que foi incluído e/ou alterado.



Nesse exemplo temos a inserção de dados de diversas tabelas que parecem ser a "stage", mas com "histórico", em uma fato de eventos.

No caso do Bancada, todos os eventos são armazenados em uma mesma fato (conformada) particionada por produtos; o mesmo acontece para Operações e Índices e Cotações.
Essas estruturas do Bancada foram implementadas para o "todo", tal qual o modelo Top-Down de Bill Inmon - algumas são fatos, outras são dimensões compartilhadas entre todos os produtos.



A denormalização de tabelas é recomendada no Hive para reduzir I/O; e no Hive reduz a quantidade de joins e, consequentemente, tasks de map e reduces.

Contudo, sabe-se que haverá sempre duplicação de algumas colunas, mas é tolerável.

Inmon - Normalização

GFT

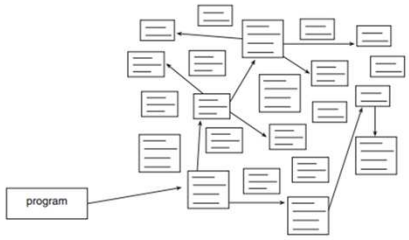


Figure 3.26 When there are many tables, much I/O is required for dynamic interconnectability.

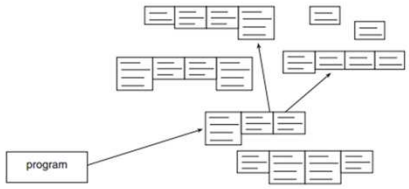


Figure 3.27 When tables are physically merged, much less I/O is required.

- Fonte: Building the DataWarehouse Third Edition – Bill Inmon

Bill Inmon também reconhece que a denormalização melhora a performance de I/O. E reduz significamente a quantidade de joins.

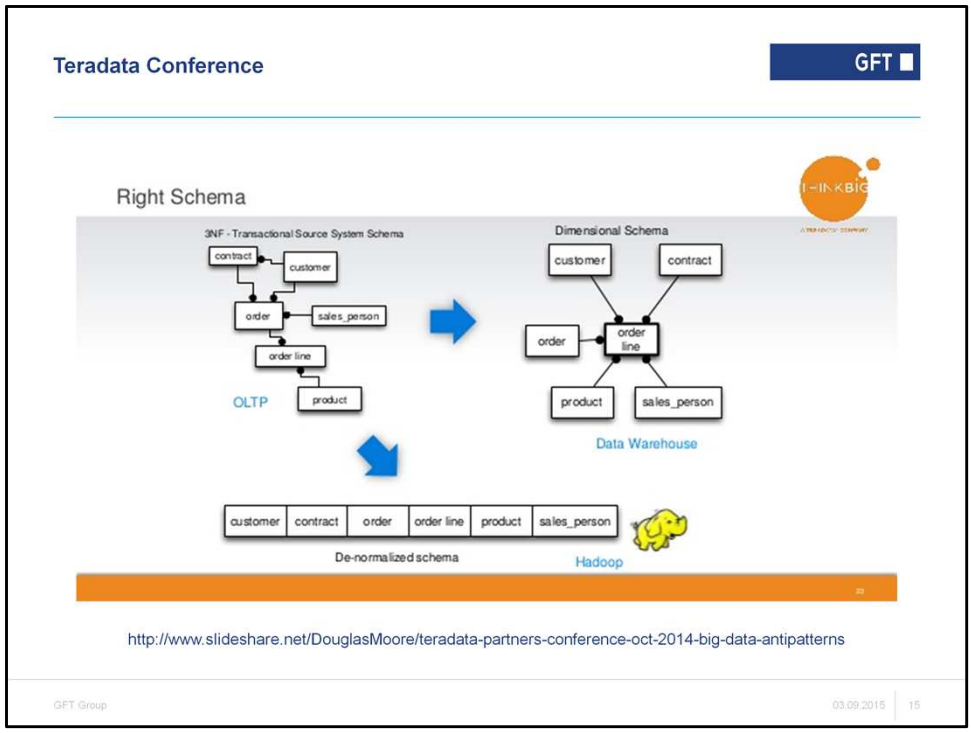
Exemplo clássico Star Schema

GFT

Figure 1.4 Fact and dimension tables in a dimensional model.

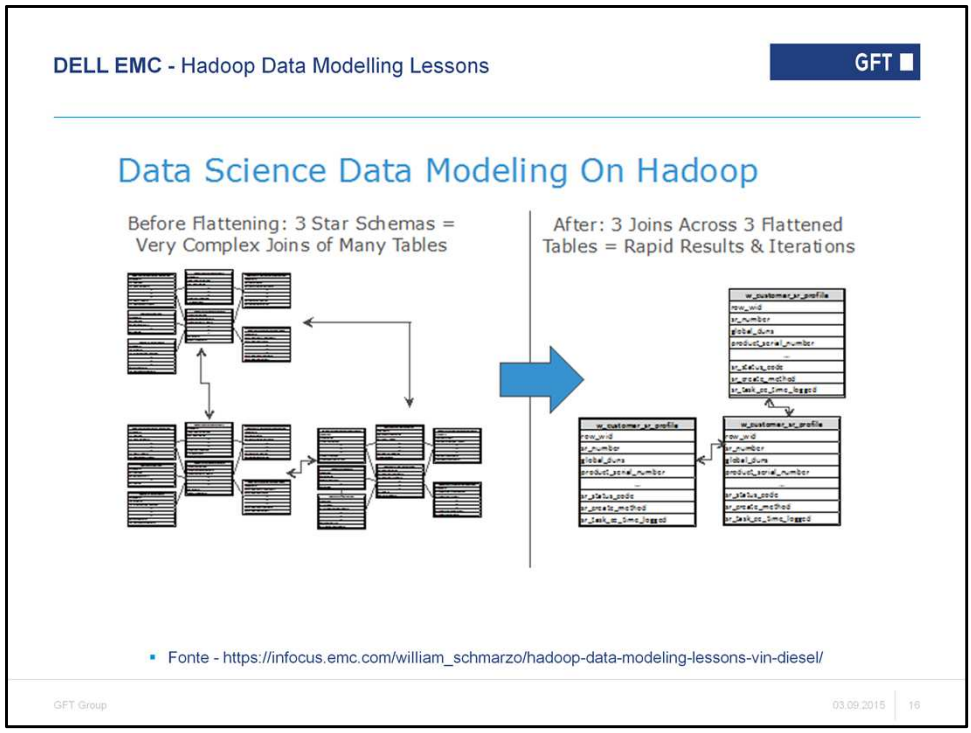
- Fonte: The Data Warehouse Toolkit Second Edition – Ralph Kimball

Exemplo clássico de Star Schema: com 3 joins é possível selecionar todos os dados de compras, seus produtos e o estoque em um determinado período.

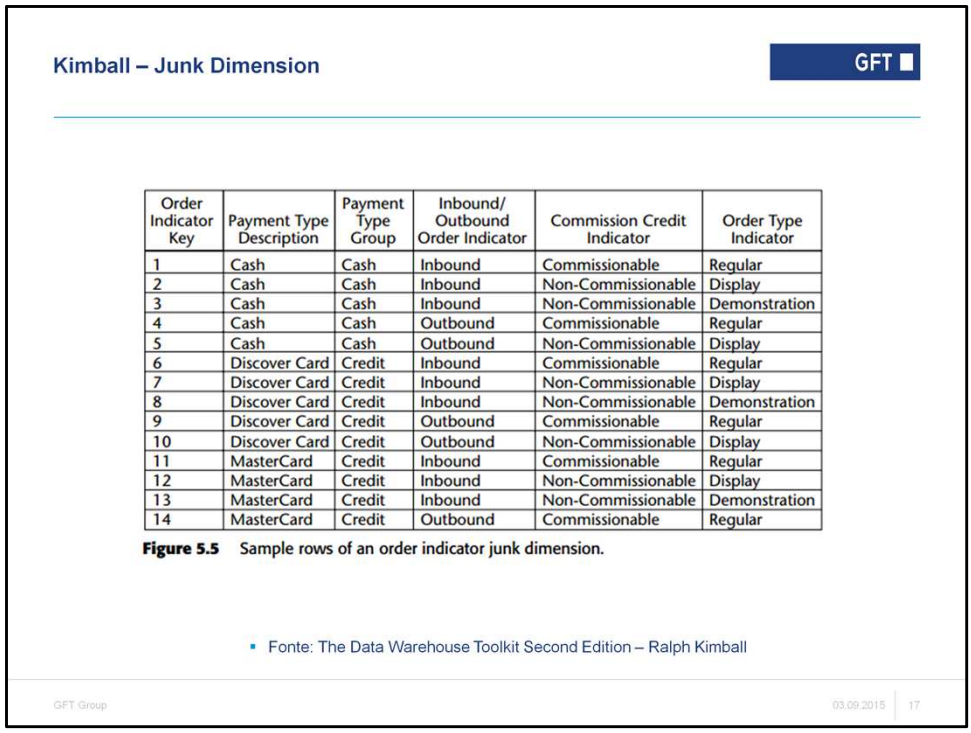


Em uma conferência da Teradata foi apresentada uma versão de schema para Hadoop em flat table, o que funciona bem para o exemplo acima.

No Bancada Analítica, algumas fatos de produtos podem possuir +200 variáveis e apenas uma coluna de valor, que pode ser atualizada todos os dias. Entendemos que não é performático, tão pouco econômico, o armazenamento nesse formato, a menos que seja extremamente necessário.



Como já pode ser constatado, não existe uma bala de prata para modelar em Hadoop, contudo todos caminham para denormalização, desde que atendam as necessidades de armazenamento e performance.



Kimball – Junk Dimension

GFT

Order Indicator Key	Payment Type Description	Payment Type Group	Inbound/ Outbound Order Indicator	Commission Credit Indicator	Order Type Indicator
1	Cash	Cash	Inbound	Commissionable	Regular
2	Cash	Cash	Inbound	Non-Commissionable	Display
3	Cash	Cash	Inbound	Non-Commissionable	Demonstration
4	Cash	Cash	Outbound	Commissionable	Regular
5	Cash	Cash	Outbound	Non-Commissionable	Display
6	Discover Card	Credit	Inbound	Commissionable	Regular
7	Discover Card	Credit	Inbound	Non-Commissionable	Display
8	Discover Card	Credit	Inbound	Non-Commissionable	Demonstration
9	Discover Card	Credit	Outbound	Commissionable	Regular
10	Discover Card	Credit	Outbound	Non-Commissionable	Display
11	MasterCard	Credit	Inbound	Commissionable	Regular
12	MasterCard	Credit	Inbound	Non-Commissionable	Display
13	MasterCard	Credit	Inbound	Non-Commissionable	Demonstration
14	MasterCard	Credit	Outbound	Commissionable	Regular

Figure 5.5 Sample rows of an order indicator junk dimension.

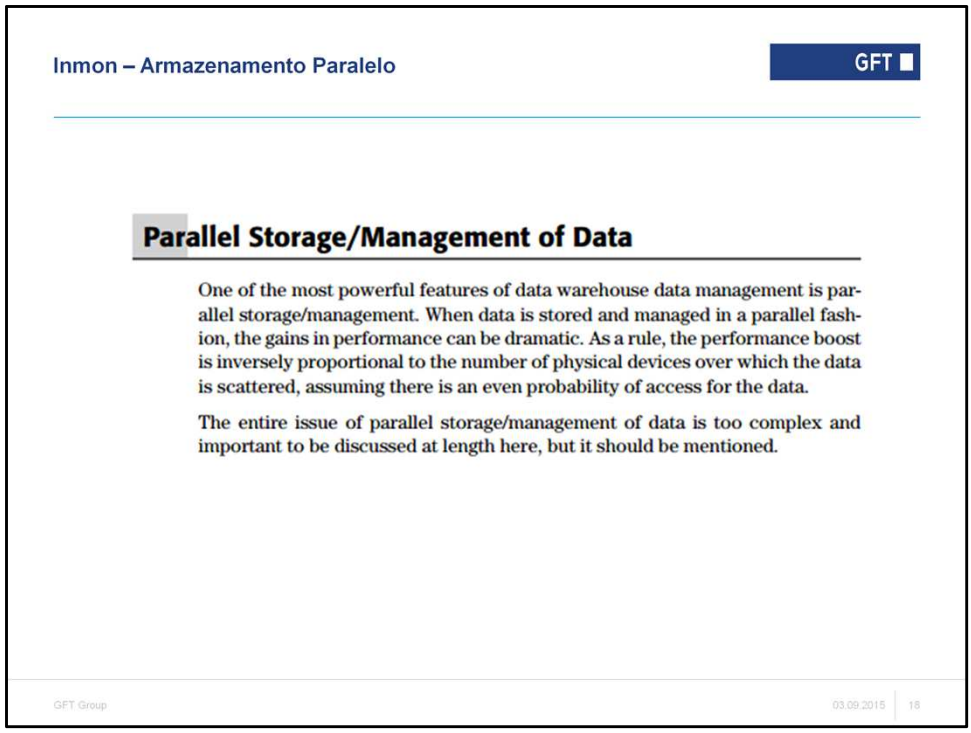
- Fonte: The Data Warehouse Toolkit Second Edition – Ralph Kimball

GFT Group 03.09.2015 17

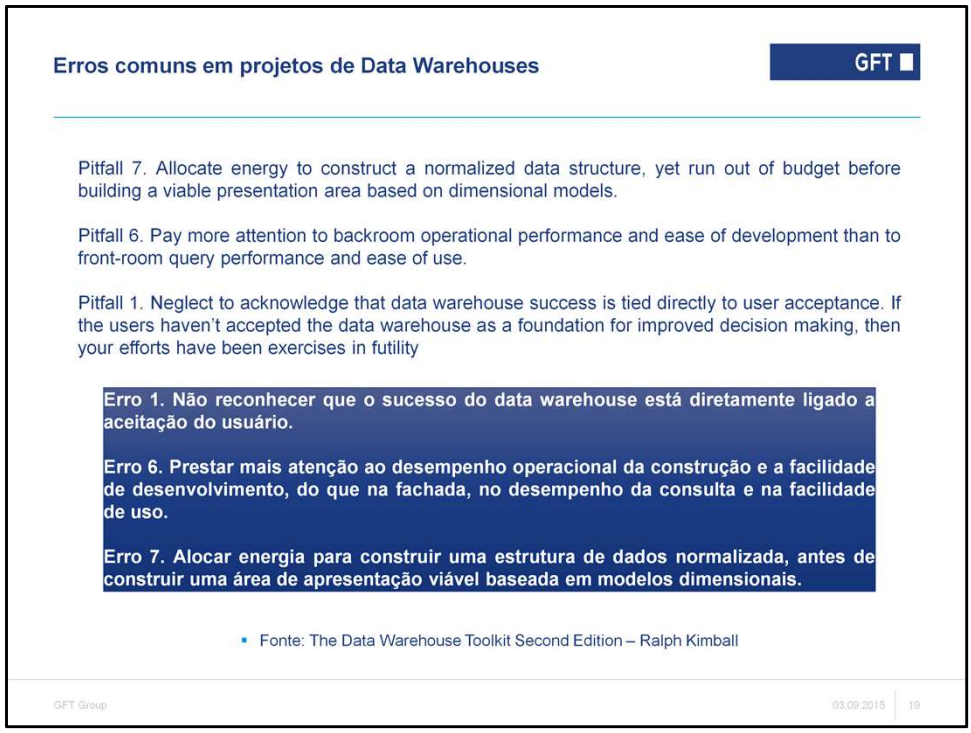
Ralph kimball propõe uma estrutura de dimensão Junk onde podemos mesclar várias dimensões em uma única dimensão, substituindo vários joins por apenas um join.

Se compararmos o modelo Star Schema com 3 joins, apresentado nos slides anteriores, unindo as dimensões em uma única "junk" teriamos apenas 1 join. Contudo, como unificar as keys de produtos, estoque e tempo?

A solução foi não receber as keys (de produtos,estoque e tempo), mas exportar a key da fato para a junk dimension utilizando assim apenas 1 join, diminuindo a quantidade de tasks, reduzindo a quantidade de incrementos e reduzindo I/O.



Bill Inmon também reconhece o poder do processamento e armazenamento em paralelo, além de complementar que realmente é complexo e necessita especial atenção.



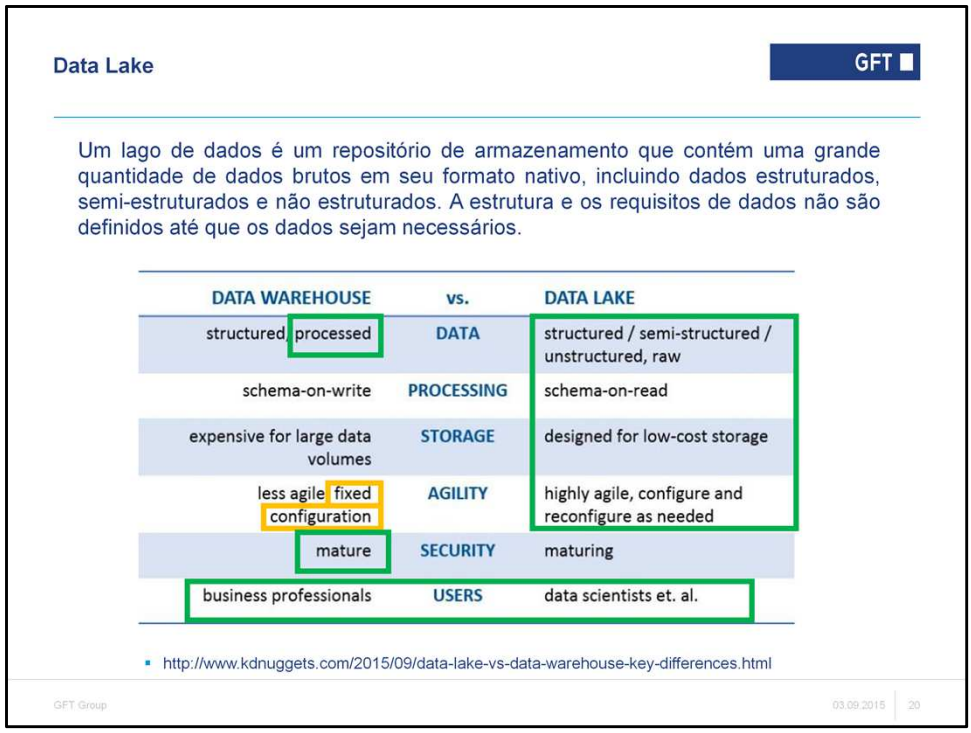
Fazer o melhor é muito importante, entregar o que foi solicitado é indiscutível!

Para usuários que em sua maioria estão acostumados a trabalhar com arquivos e seus processos internos, mudar drasticamente todos os processos internos de cada área é com certeza um risco muito grande de rejeição ao projeto.

É preciso entregar o que eles necessitam e garantir sustentação aos processos já existentes, em algum ponto eles irão evoluir de usuários consumidores a analistas e perceberão que o ambiente e a ferramenta não são mais sustentáveis e migrarão seus processos aos poucos para o Bancada.

Gastar esforços para construir uma estrutura "a mais" tem impactos:

1. Maior quantidade de scripts a serem desenvolvidos;
2. Maior complexidade do processo;
3. Duplicidade de informações em diferentes estágios;
4. Maior tempo de execução da malha de carga;
5. Manutenção mais difícil.



Quando compamos o Bancada Analítica com os conceitos de Data Warehouse e Data Lake podemos verificar que ele possui caraterísticas de ambos os lados.

O Bancada atualmente só recebe conteúdo estruturado, mas existem requisitos para armazenamento de arquivos não estruturados.

Todas as tabelas são “external tables“ e possuem schema on read, o que facilita a alteração de estrutura e tipo de dado.

Os dados do DWH e do DataMart são processados através de uma malha de carga.

O Bancada foi concebido como um repositório único de acesso a dados para todos os tipos de profissionais, excetuando-se aqueles com restrições de segurança.

Qual a Metodologia aplicada ao Bancada ?

GFT

- Modelo híbrido top-down e botton-up
  - Algumas dimensões e fatos são pensados Top-Down (Cadastros, Índices, Operação, Eventos), pois são compartilhados.
- Implementado por Produtos
  - Bottom-up.
- Stage
  - Cópia fiel da única origem (Oracle);
  - Informações substituídas todos os dias.
- DWH ou EDW
  - Área de armazenamento dos DataMarts (Produtos) e dimensões primárias compartilhadas;
  - Snow Flake de fatos e dimensões normalizadas;
  - Informações incrementadas todos os dias.
- DataMarts
  - Flat Tables para atender requisitos específicos dos usuários / áreas;
  - Formato escolhido para minimizar joins;
  - Mantém Surrogate Keys para relacionamento com DWH, caso haja necessidade de mais informações ou rastreabilidade;
  - Fotos diárias das informações.

GFT