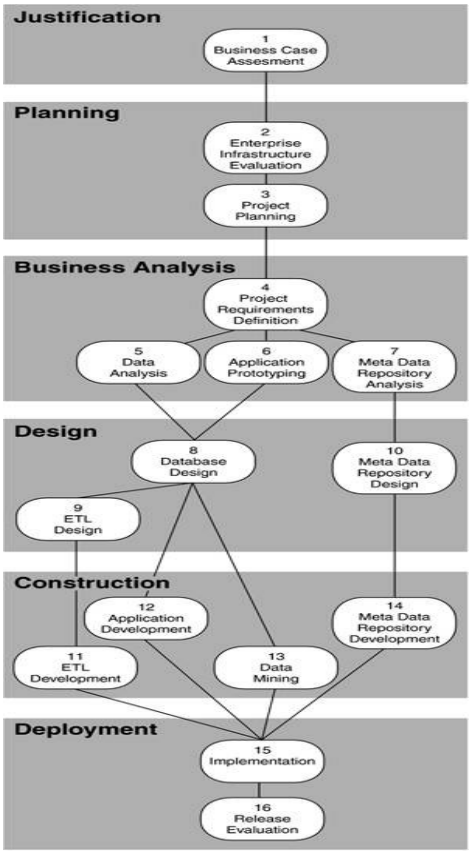
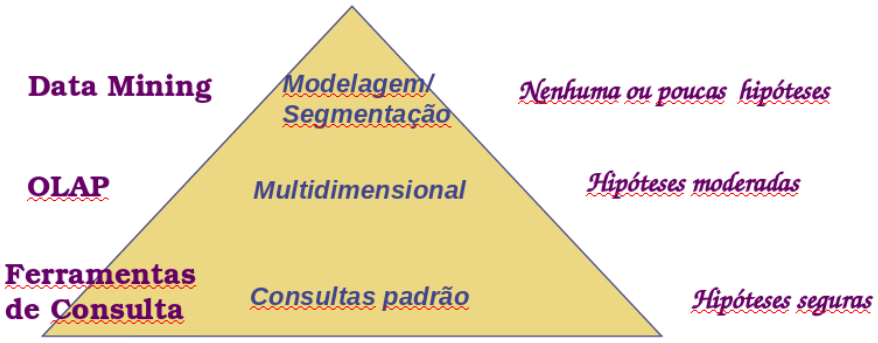


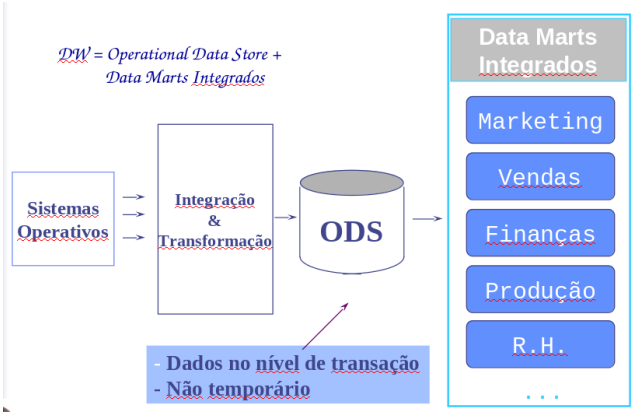
Drawing 1: Etapas do projeto de um DW

- **Data Warehouse:** ambiente para organizar, gerenciar e disponibilizar informações oriundas de fontes diversas (sem impacto para o ambiente operacional de dados da empresa), fornecendo uma visão única de parte ou de todo o negócio com o objetivo de dar suporte a operações analíticas (como tomadas de decisão.)
- Propõe-se a dar a informação de qualidade com agilidade, flexibilidade, e em uma única versão de verdade (setores como financeiro e RH podem ter diferentes versões de verdade em seus discursos, por exemplo: se se perguntar quanto se gasta com o salário, o RH pensará somente no de seus funcionários, enquanto que o financeiro pensará também em seus terceirizados e assim vai; situação que os sistemas tentam evitar.)

- **Processamento OLTP:** é do ambiente operacional (estoque, expedição, administrativo), baseado em transações. Preocupa-se em executar de forma mais rápida atividades repetitivas, situações correntes (como um processamento de folha de pagamento), trabalhando, para isto, com alto nível de detalhe, sendo suas atualizações e consultas em grande nível.
- **Processamento OLAP:** é do ambiente analítico (suporte à decisão, auxilia na concepção negocial da empresa), baseado em um número limitado de consultas variáveis, *ad-hoc*, feitas conforme necessárias e, geralmente, complexas. Suas aplicações são dinâmicas: o dado é visto sob diferentes perspectivas. Utiliza muitas operações de agregação e cruzamento. Dificilmente executa atualizações, apenas inserções (novos *logs* e conhecimentos, p. ex.) e muitas seleções. Naturalmente, dados históricos são muito importantes, e consistência é fundamental.
- **Sistemas Operacionais:** preocupam-se com tempo de resposta, a segurança, a recuperação de falhas, e com acesso concorrente de usuários.
- **Sistemas Analíticos/Informacionais:** preocupam-se com flexibilidade de navegação, isto enquanto gerenciam enorme volume de dados e metadados, os quais devem ser examinados em muitos níveis de detalhe, e oriundos de fontes diversas.



- Os data warehouse podem ser de dois tipos:
  - **Setoriais:** são evolutivos, focando inicialmente apenas nos aspectos mais críticos, o que garante um retorno rápido enquanto fornece acúmulo de experiência, provendo, assim, menor custo e risco. Outra vantagem é aproveitarem a estrutura institucional disponível. São chamados de **Data Mart** os setores. Em uma **abordagem bottom-up**, um DW é composto pelos vários data marts.
  - **Corporativos:** de grande abrangência, manipulando dados de todos os setores, portanto muito complexos e com alta probabilidade de insucesso. Em uma abordagem **top down**, data marts representam uma versão sumariada do DW corporativo e, portanto, só podem ser concebidas ou pensadas após se ter desenvolvido o DW completo e centralizado.



- Atualmente, prega-se uma visão integrada, a fim de errar o mínimo possível. Com isto, o planejamento ocorre de maneira top-down, mas o desenvolvimento se dá de forma bottom-up, trabalhando-se um Data Mart por vez, com resultados atingidos em pequenos ciclos. Na figura ao lado, o ODS (Operational Data Store) funciona mais ou menos como um ambiente temporário de dados detalhados, o qual é o nível atômico, uma *view* comum, do DW.
- Inmon aponta modelos entidade-relacionais como úteis ao DW, permitindo consistência e flexibilidade. Acredita, também, que as formas normais são a base do *design* do DW e do Operational Data Store (ODS.) Só não são ideais ao Data Mart.
- Kimball não concorda com estes modelos, por serem complicados para usuários-comuns. Formas normais não são úteis para OLAP, apenas OLTP, pois eliminam redundâncias, essenciais à performance. Isto se deve ao fato de que modelos entidade-relacionais são otimizados para performance de atualizações, e não de seleções.

- **Transporte de Dados/Data Staging**: mesma coisa que ferramentas ETL:
  - **Extração:** coleta os dados, uma operação demorada e complexa; muitas vezes *ad-hoc*.
  - **Transformação:** útil para a clareza e integração, recodificando categorias (m/f para masculino/feminino, p. ex.), uniformizando e alterando unidades de medidas, nomes de campos, datas.
  - **Limpeza:** melhora a qualidade da informação extraída.
  - **Carga (load):** se cargas são muito frequentes, é caro; se são poucas, trabalha-se com dados muito velhos.
- Ferramentas ETL geralmente desenvolvidas invés de compradas. A maioria dos produtos inclui transformadores proprietários, ou geradores de código. São ideais para ambientes complexos.

Componentes Potenciais de um DW

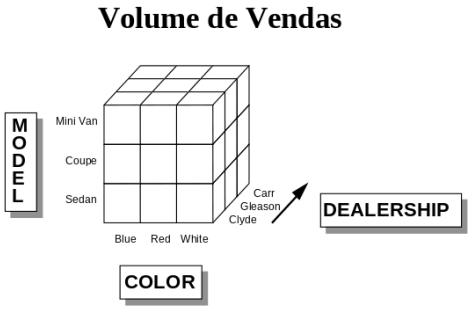
- Repositório de metadados (todos os componentes geram metadados.)
- Bancos de dados relacionais, não-relacionais e multidimensionais.
- Provedores de interfaces (ODBC/OLE) e gateways (para BDs legados.)
- Ferramentas: ETL; para qualidade e limpeza; de replicação; OLAP; de relatório e consulta; de data mining; de monitoramento e controle; de agendamento *batch* multi-plataforma.
- Ferramentas de projeto CASE (auxílio para atividades de engenharia de software.)
- Pacotes de aplicações para Data Warehouse.

Metadados

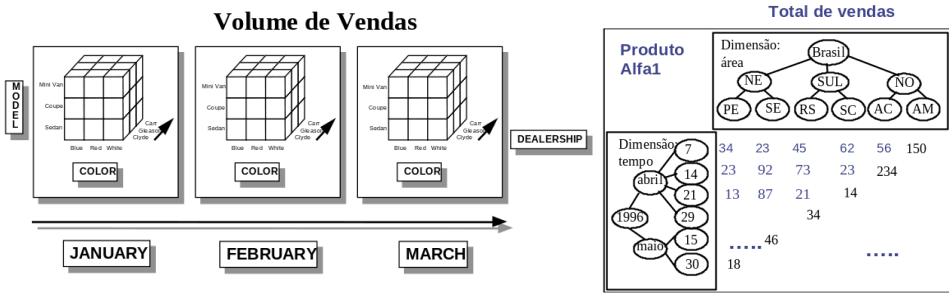
- Úteis para quando os formatos de dados são inconsistentes, ou quando estes dados são mesmo inexistentes ou inválidos. Permitem lidar também com inconsistências semânticas, trabalhando-se com qualidade de dados e com janelas de tempo. Deve-se ter, a eles, acesso global, além de ser possível fazer administração e controle deles. Metadados possuem diferentes níveis de agregação.
- Metadados podem ser de dois tipos:
  - **Técnicos:** altamente estruturados, informando definições, transformações, gerências e operações. São geralmente tratáveis com ferramentas de repositório.
  - **De Negócio:** ambos estruturados e não-estruturados, são mais difíceis de serem tratados e integrados por ferramentas muito estruturadas como as de repositórios.
- **Repositórios:** proveem armazenamento, acesso e gerência (até mesmo do ciclo de vida) de metadados, oferecendo uma visão global e integrada destes.

MODELAGEM DE DW

- Como as medidas de análise devem ser vistas sob diferentes perspectivas, a abordagem utilizada é a de modelagem dimensional. Apesar do nome foda, o conceito em si é meio meh.



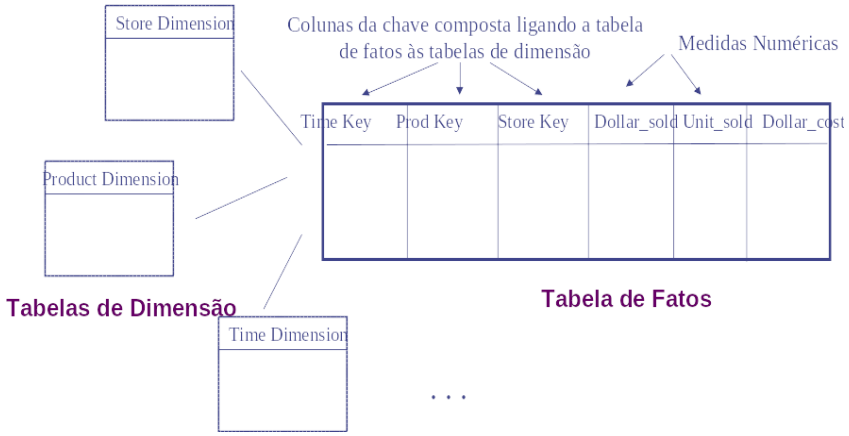
- Invés de exibir os dados em formatos tabulares, pode exibi-los de forma dimensional (isto é, altura x largura, como uma matriz, ou mesmo como um cubo, ou bizarrices outras acima de três dimensões.) Este cubo é apenas uma metáfora visual; uma representação intuitiva dos dados.



- Os dados são também altamente hierarquizados, sendo estas hierarquias a base das agregações.
- Um dos grandes problemas dos agregados é saber se devemos calculá-los no momento de recuperação ou se devemos já os deixar previamente armazenados. Na prática, armazena-se apenas parte dos agregados, e há tratamento para dados esparsos.
- Deve-se, também, tomar cuidado porque, quanto maior o número de dimensões, maior o número de combinações possíveis de agregados (8 dimensões daria 2^{2*8}=65536 agregados.)

O Esquema Estrela

- Distingue melhor as dimensões dos fatos medidos, simplificando a forma dimensional de visualização. Na prática, mistura modelagem conceitual com modelagem lógica, pois é muito voltada para a abordagem relacional (especialmente tabelas.)
- Trata-se, basicamente, de uma tabela de fatos, cercada por várias tabelas de dimensões. Estas tabelas dimensionais são ligadas por *join* à de fatos.
- Nesta tabela de fatos, ficam guardadas, além dos fatos desejados, as *surrogate keys* das tabelas das dimensões. Geralmente, estas dimensões são: *o quê?*, *onde?*, *quem?* e *quando?*
- Se comparado a um modelo entidade-relacional normal, o esquema estrela é assimétrico.



- A tabela de fatos é dominante, com grande volume de dados. Sua chave é composta, e o tempo é sempre parte da chave. É usualmente apenas numérica. Fatos são, tipicamente, aditivos.
- As dimensões nada mais são do que tabelas as quais qualificam os fatos. Costuma ser de menor volume que a tabela de fatos. Possuem uma chave simples, servindo como cabeçalho das linhas e colunas das análises; é um filtro, nas consultas. Para cada registro, a descrição deve ser única. Costumam não depender de tempo. Tem hierarquias implícitas. São tabelas desnormalizadas. Dimensões podem ter múltiplas hierarquias, além de outros atributos descritivos. Por exemplo, a geografia física, para um atacadista, pode ser cidade, estado, região, país etc. A de vendas, território, região, zona. A de distribuição, AD primária, região etc.
- Uma variação do esquema estrela é o **esquema floco-de-nove**, em que as hierarquias são explicitadas por meio de uma normalização na tabela das dimensões. Assim, se temos uma dimensão "Armazém" que contém a chave e a cidade de um armazém, esta pode estar ligada a uma outra tabela, "Cidade", que contém o nome da cidade e um estado, esta estando ligada a outra tabela, "Estado", que contém o nome do estado e um país e assim sucessivamente.

OU, LE WEB…

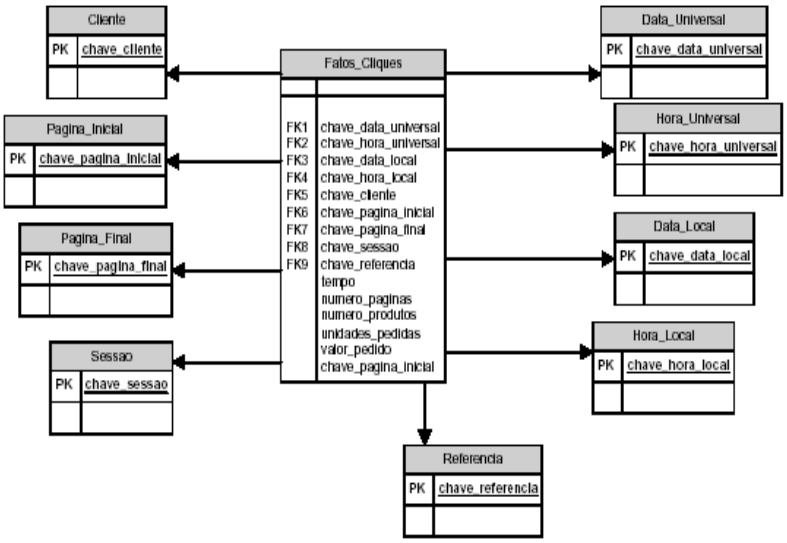
- Antes, a Web era vista como uma forma de disponibilizar informação ou sistemas. Hoje, ela é vista como um grande banco de dados ~~só pra roubar de forma antiética suas informações privês.~~
- Um dos problemas nesta abordagem é que o porte da web é amplo demais para DW/data marts efetivos: há muita informação sem utilidade nem qualidade, além da falta de estrutura e da dinamicidade das informações, que são atualizadas constantemente, havendo cobertura limitada (muitos dados escondidos.)
- Assim, há diferentes abordagens para explorar a web: por conteúdo, de estrutura, de utilização.

Exploração de conteúdo

- A mineração de dados serve para lidar com alguns destes problemas. Sua cobertura é estendida para depois se encolher, usando sinônimos e hierarquias de conceito. Suas primitivas de busca costumam ser dicas ou preferências do usuário. Além disso, permite fazer análise de ligações.
- Para uso de metadados, o XML pode ajudar, mas a liberdade de criar tags dificulta a integração de informações mais gerais (por exemplo: o atributo *title* é o título de uma página HTML ou é um título aleatório de um XML?)

Exploração de log

- A mineração também pode ser feita em cima de *logs* de servidores web. Assim, pode-se analisar séries de tempo, associações, classificações. São úteis para compreender o comportamento com relação à estrutura do *site*, contribuindo no projeto destes, visando clientes potenciais. Pode-se, p. ex., analisar estatísticas de completude e tempo da visita, número de páginas visitadas, páginas mais visitadas, quantidade de erros, análise de performance e tráfego etc.
- *Logs* são úteis porque recriam o comportamento de um ambiente real, sendo coletados de forma desapercebida, baseados em comportamentos observáveis (e não em relatórios preenchidos pelo usuário), ordenados, por padrão, de forma temporal, e sem perturbações de agentes de pesquisa.
- Alguns dos problemas são que clientes podem gerar muitas conexões, ou acessar páginas de máquinas distintas, ou ter-se um cliente usando vários IPs ou vice-versa, apenas um IP sendo usado por vários clientes, devido a máquinas proxy. As páginas podem vir do cache, o que não conta para o log. Igualmente, é difícil estabelecer o fim de uma sessão, bem como botões de ir e voltar do navegador confundem o log. Bots/spiders/agentes geram requisições automáticas, o que dificulta diferenciá-los de clientes. Se a página for dinâmica, será impossível reconstruir, somente pelo log, aquilo que um cliente viu. O timeout é outro problema, por seus arbitrários 30 minutos. Por fim, requisições POST são outro problema, uma vez que elas não são registradas no log ~~grazadeus né porque também pra que segurança???~~
- Pode-se, no entanto, fazer um pré-processamento antes da mineração de *logs* para que se possa identificar usuários, associar uma sessão a outros dados e preencher dados que sejam incompletos ou inexistentes, bem como reconstruir o *clickstream* (rastro de cliques.) Problemas específicos de data marts de *clickstreams* é o volume monstruoso de dados e as cargas frequentes que precisam ser realizadas.



- Com isto, pode-se moldar um data mart cuja tabela de fatos seja uma sessão, com as dimensões: data (local e universal), horário (local e universal), cliente, página de entrada, sessão e referência. Suas medidas podem ser duração da sessão em segundos, páginas visitadas, pedidos feitos, unidades pedidas e valor pedido.