FIAP

Sistemas de informação

enterprise analytics and data warehousing

Profº Fabiano J. Cury Marques

FIAP

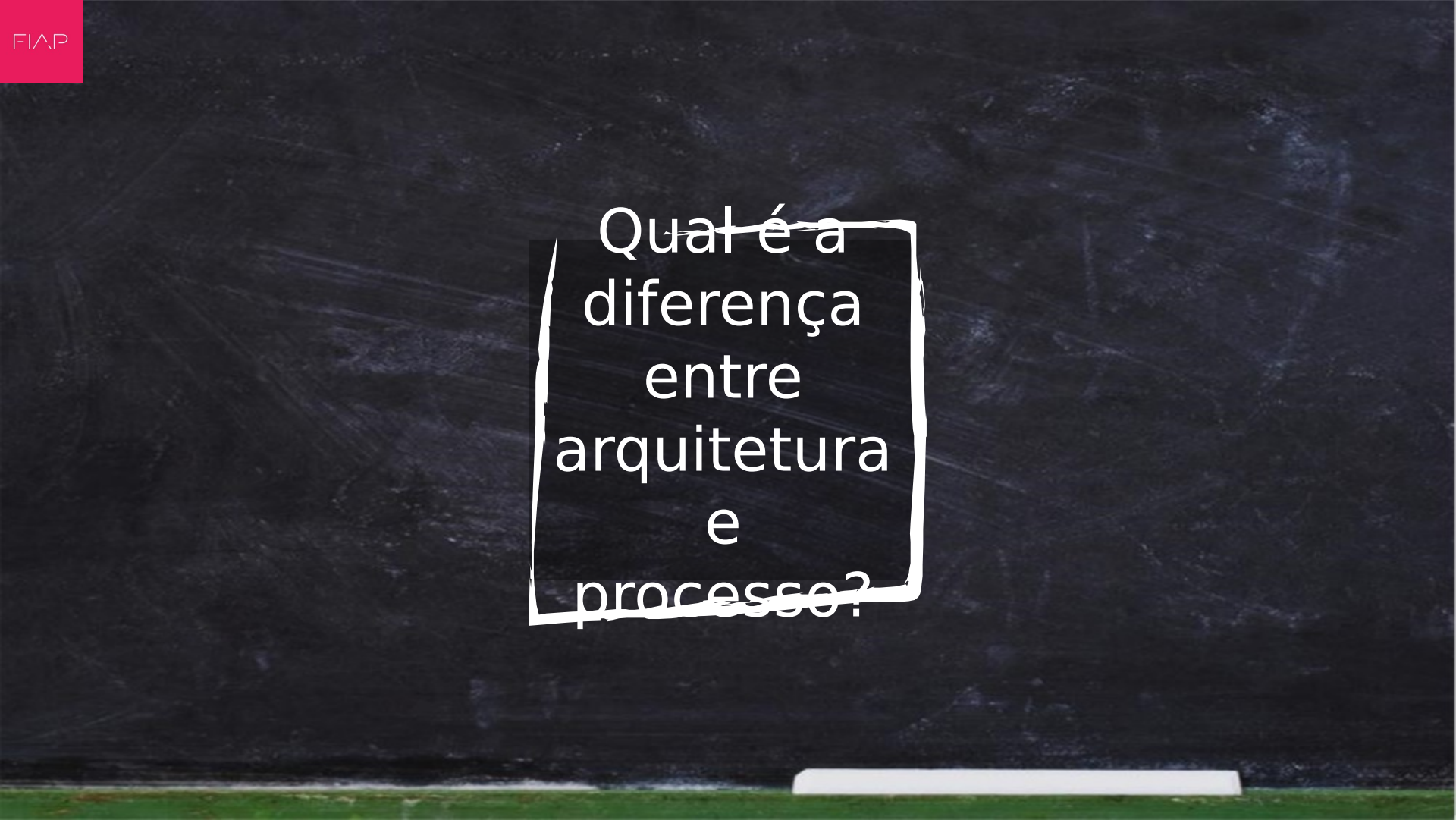
enterprise analytics and data warehousing

Agenda

- ✘ Arquitetura DW
- ✘ Arquitetura de Fluxo de Dados
- ✘ Arquitetura de Sistemas
- ✘ Referências

Arquitetura de DW

Arquitetura de DW


Qual é a diferença entre arquitetura e processo?

Introdução

Um DW tem duas arquiteturas principais que devem ser notadas:

- Arquitetura do fluxo de dados
  - Como os repositórios de dados são arranjados em um DW e como os dados fluem a partir dos sistemas-fontes até os usuários através destes repositórios;

- Arquitetura do sistema
  - Configuração física dos servidores, rede, software, storage e clientes;

Arquitetura do fluxo de dados

- A arquitetura do fluxo de dados é uma configuração dos repositórios dentro do DW;
- Isto inclui como o fluxo de dados é controlado, logado, e monitorado, assim como o mecanismo para garantir qualidade dos dados nos repositórios;
- Vamos discutir três arquiteturas:
  - DDS
  - NDS + DDS
  - ODS + DDS
- Não deve-se confundir com arquitetura de dados
  - Que refere-se a **modelagem de dados**

Repositórios

- Os repositórios são componentes importantes da arquitetura de fluxo de dados;
- É uma ou mais bases de dados ou arquivos contendo dados do DW, arranjadas em um formato específico e envolvidas em processos do DW;
- Baseado em acessibilidade do usuário, pode-se classificar em :
  - **User-facing**: disponível para usuários e suas aplicações e queries;
  - Interno: usado internamente pelos componentes do DW com o propósito de integrar, limpar, “logar”, e preparar dados. Não é aberto para os usuários;
  - Híbrido: utiliza os conceitos user-facing e interno;

Repositórios

- Baseado em formato de dados, pode-se classificar em:
  - **Stage:** **repositório interno** usado para transformar e preparar os dados obtidos a partir dos sistemas-fontes, antes dos dados serem carregados para outros repositórios no DW;
  - **Normalized data store (NDS):** **repositório interno** na forma de uma ou mais bases de dados relacionais normalizadas com o objetivo de integrar dados de várias fontes capturadas em uma stage, antes dos dados serem carregados em um repositório user-facing;
  - **Operational data store (ODS):** **repositório híbrido** na forma de uma ou mais bases de dados relacionais normalizadas contendo os dados de transações e a versão mais recente dos dados principais, com o objetivo de suportar aplicações operacionais;
  - **Dimensional data store (DDS)**: **repositório user-facing** na forma de uma ou mais bases de dados relacionais, onde os dados são arranjados em um formato dimensional com o objetivo de suportar consultas analíticas;

Termos importantes

- Base de dados relacional
  - Consiste de tabelas de entidades com relacionamentos pai-filho entre elas
- Base de dados normalizada
  - Base de dados com pouca ou nenhuma redundância de dados que está ao menos na terceira forma normal
- Base de dados desnormalizada
  - Base de dados com alguma redundância de dados que não passou pelo processo de normalização
- Base de dados dimensional
  - Base de dados desnormalizada que consistem de tabelas fatos e dimensões

Termos importantes

- MDB
  - Algumas aplicações necessitam que os dados estejam na forma de uma base de dados multidimensional (MDB) em vez de relacional
  - MDB é uma forma de base de dados onde os dados são armazenados em células e a posição de cada célula é definida por variáveis
  - Cada célula representa um evento do negócio e o valor das dimensões mostram quando e onde o evento ocorreu
  - MDB é populado a partir do DDS
- ETL
  - Extract, Transform, and Load (ETL) é um sistema que tem capacidade de ler os dados de um repositório, transformar os dados, e carregá-los em outro repositório
  - O mecanismo para logar o resultado de cada passo do ETL é chamado de auditoria do ETL
  - A descrição de cada processo do ETL é armazenada nos metadados. Isto inclui dados da fonte, destino, as transformações aplicadas, o agendamento de processos etc.

Termos importantes

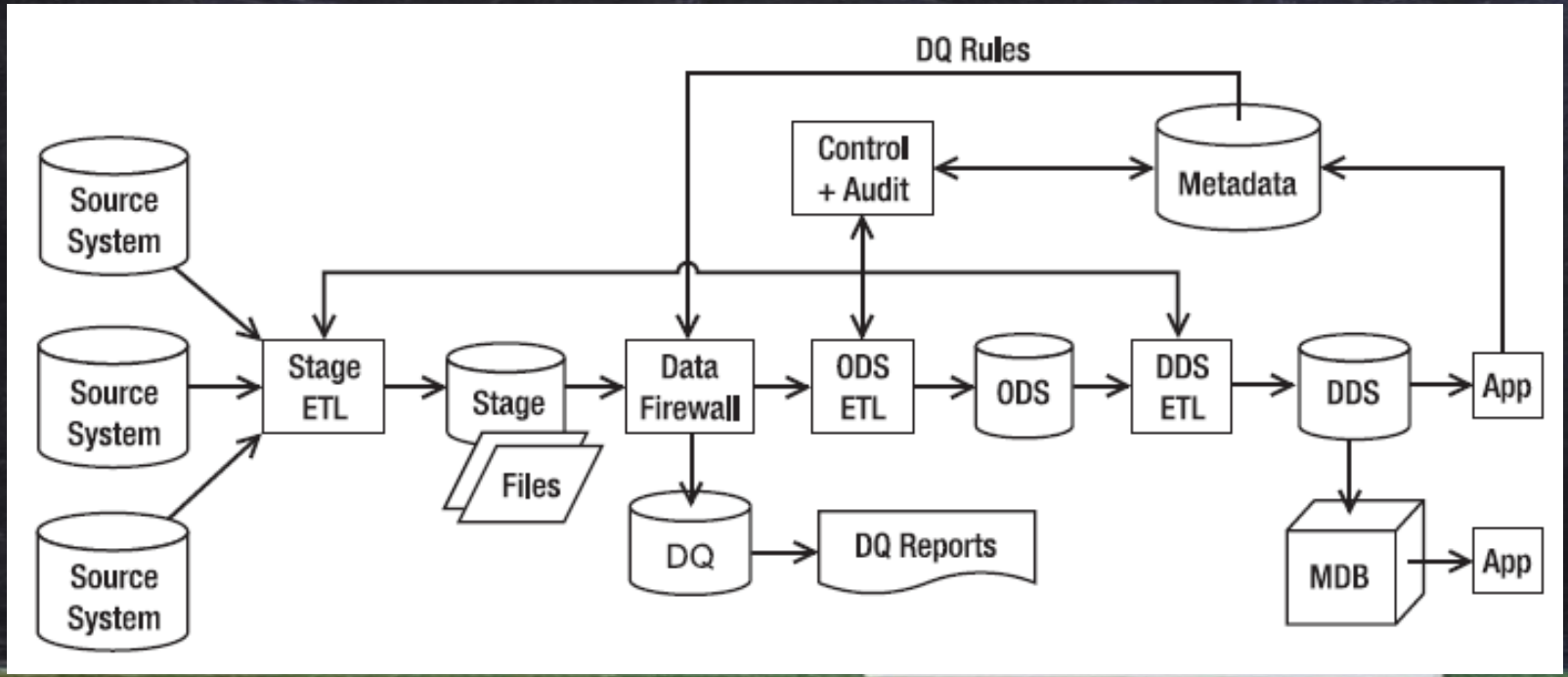
- **Data Quality**
  - São atividades e mecanismos para garantir que os dados no DW estão corretos e completos
  - Também cobre o mecanismo de reportar os dados incorretos / incompletos e corrigi-los
  - Um data firewall é um programa que verifica se o dado que está sendo incluído atende a todas as regras de qualidade
  - Base de dados de Data Quality contém dados que não atenderam as regras de qualidade

Atividade

- **“Veste Bem S/A” é uma empresa que tem 500 lojas franquias no estado de são Paulo. A fabrica da empresa produz calçados para o público feminino, fornecendo todos os produtos para as lojas franquiadas. A empresa trabalha com 5 canais de venda (balcão, delivery, web, aplicativo e mercado livre), sendo que cada um deles está relacionado a um sistema transacional diferente. O CEO da empresa, pretendendo investigar em mais detalhes o processo de vendas, determinou que um DW deve ser implantado. A pedido do CEO, alguns analistas de negócio se reuniram com a equipe de TI para definir as diretrizes iniciais de um DW. Desenhe o esboço de uma arquitetura de fluxo de dados (repositórios e processos/serviços), para atender a necessidade da empresa.**

Arquitetura Completa

A figura abaixo demonstra uma possível arquitetura com todos os componentes como sistema de controle, metadados e componentes de qualidade de dados



Importância

1. A **arquitetura do fluxo de dados** é **uma das primeiras coisas** que deve ser decidida na construção de um DW
2. Determina quais componentes precisam ser construídos e portanto afetam o **planejamento do projeto** e os custos
3. O fluxo de dados é projetado de acordo com os requisitos de dados das aplicações. Aplicações de DW precisa de dados em diversos formatos
4. Depois de determinar os repositórios necessários para o DW, pode-se **projetar o ETL para popular** estes repositórios
5. Enfim, define-se o mecanismo de qualidade de dados

Arquitetura 1: DDS Simples

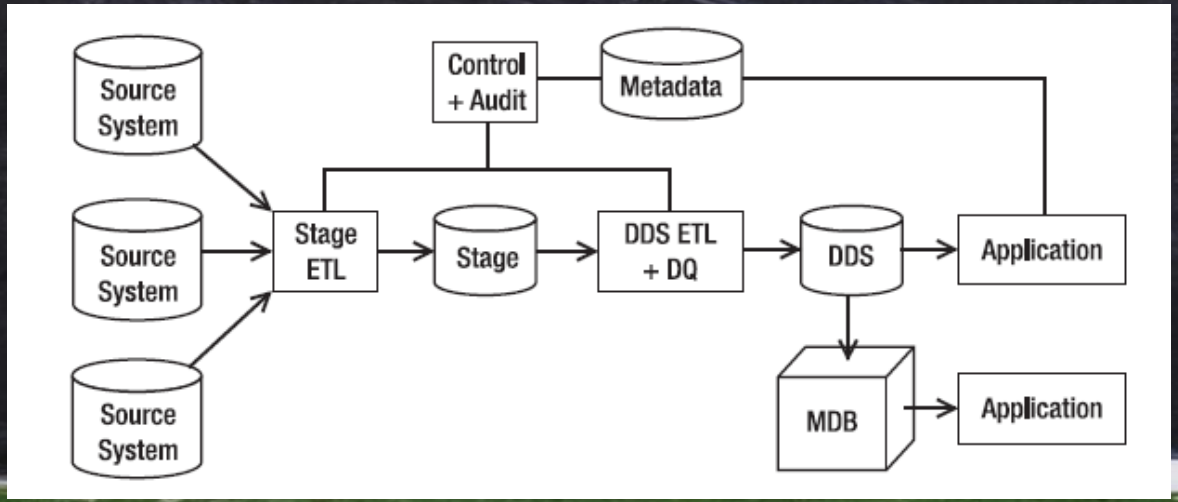
- **Arquitetura mais simples de todas**
- **Consiste de apenas dois repositórios**
  - **Stage**
  - **DDS**
- **Nesta arquitetura se tem apenas um repositório dimensional**
  - **DDS contém um ou mais data marts dimensionais**
  - **Um data mart dimensional é um grupo de tabelas fatos relacionadas e suas tabelas dimensões correspondentes**
- **Um pacote de ETL extrai dados de diferentes fontes e os coloca na stage**

STAGE

- É o lugar onde se colocam os dados extraídos das fontes temporariamente, antes de os processar
- É necessária quando as transformações são complexas (ou seja, não podem ser feitas "*on the fly*"), quando o volume é grande ou quando os dados de diversas fontes chegam em diferentes momentos
- Também necessária quando se deseja minimizar o tempo para extrair os dados da fonte
- O ETL que extrai dados das fontes os insere em uma base de dados (stage). Um segundo ETL pega os dados da stage, integra os dados de diversas fontes, aplica critérios de qualidade e os coloca consolidados no DDS

Arquitetura 1: DDS Simples

- Control + Audit refere-se ao sistema que **controla e audita o ETL**
- A base de dados de **metadados** contém as **descrições de estruturas**, **dados**, e **processos** dentro de um DW
- As aplicações do DW, como ferramentas de relatórios BI, lêem os dados no DDS e trazem para os usuários
- Os dados no DDS podem também ser carregados para MDB, como o **SQL Server Analysis Services** e acessados pelos usuários via aplicações **OLAP** e **Data Mining**



Arquitetura 1: DDS Simples

- **Vantagens**
  - **É a mais simples das arquiteturas possíveis para um DW**
  - **Isso porque os dados da stage são carregados diretamente no repositório dimensional, sem passar por nenhum processo de normalização antes**
  - **Construção mais simples e ETL mais rápidos**

- **Desvantagens**
  - **É mais difícil nesta arquitetura criar um segundo DDS**
  - **O DDS nesta arquitetura é o repositório principal, ele contém um conjunto completo de dados no DW incluindo todas as versões e todo o dado histórico**
  - **As vezes, necessita-se criar DDS menores contendo apenas um subconjunto de dados do repositório principal para um objetivo específico de análise**
  - **Neste caso, seria necessário criar um novo ETL para ler os dados da DDS principal e carregar os DDS menores**
  - **Não há como reaproveitas os ETL existentes**

Arquitetura 1: DDS Simples

- **Quando usar**
  - **Usa-se quando se necessita de apenas um repositório dimensional e não precisa-se de um repositório normalizado**
  - **Usado em soluções de BI simples e rápidas onde os dados são utilizados apenas para alimentar um DW dimensional**
  - **Esta arquitetura é particularmente aplicável quando se tem apenas uma fonte de dados pois não é necessário repositórios adicionais (NDS ou ODS) para integrar os dados de diferentes fontes**

Arquitetura 2: NDS + DDS

- **Nesta arquitetura existem três repositórios:**
  - **Stage**
  - **NDS**
  - **DDS**

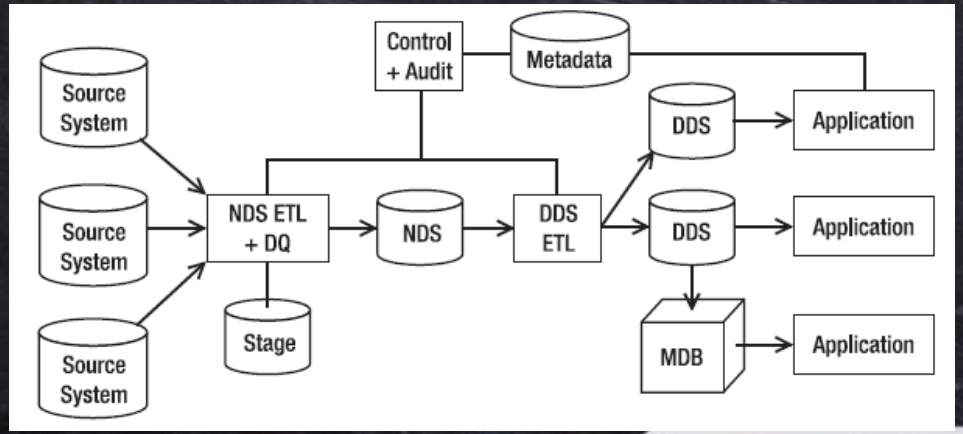
- **É similar a arquitetura DDS simples, mas tem um repositório normalizado antes do DDS**

- **NDS tem dois objetivos básicos:**
  - **Integrar dados de diversas fontes diferentes**
  - **Permitir carregar dados em diversos DDS**

- **Ao contrário do DDS simples, no NDS + DDS pode-se ter diversos DDS facilmente**

Arquitetura 2: NDS + DDS

- Nesta arquitetura, **NDS é o repositório principal**, o que significa que ele contém os conjuntos de dados completos, incluindo todo o histórico de transações e todas as versões de master data.
- O **DDS por sua vez não é o repositório principal**. Ele pode não conter todos os dados de transações
- **NDS contém todas as versões históricas** de master data. Se houver uma mudança, os atributos não são atualizados, e sim inclui-se **um novo registro e a versão antiga é mantida na mesma tabela**



NDS

- Similar aos sistemas transacionais (OLTP), existem dois tipos de tabelas em um NDS
  - **Tabelas de transação**: contém uma transação de negócio ou evento de negócio. Exemplo: **tabela de pedidos de venda**
  - **Tabelas master**: contém as pessoas ou objetos envolvidos em um evento de negócio. Exemplo: **tabela de produtos**

- As **tabelas fatos do DDS** são carregadas a partir das **tabelas de transação da NDS** e as **tabelas dimensões do DDS** são carregadas a partir das **tabelas master da NDS**

- NDS é um repositório interno, ou seja, não é acessível pelo usuário final

- A única aplicação capaz de atualizar o NDS é o ETL do NDS

Arquitetura 2: NDS + DDS

- **O ETL para carregar a DDS é mais simples que na arquitetura anterior**

- **Pode-se criar diversos DDS a partir do NDS com um mesmo ETL**
  - **Para isso, deve-se deixar o programa de ETL parametrizável em relação a tabelas, colunas e linhas que deseja-se transferir**
  - **Por exemplo, pode-se criar um DDS contendo apenas um data mart de rentabilidade de clientes que contém apenas os últimos três meses de dados**

- **Em NDS + DDS pode-se ter diversos DDS, porém um é obrigatório: aquele que contém todas as tabelas fatos e dimensões. Todos os outros são opcionais.**

- **Nas tabelas NDS ficam as surrogate keys e as natural keys**

Arquitetura 2: NDS + DDS

✘ Vantagens

- Possibilidade de recriar facilmente o DDS principal
- Facilmente se cria um novo DDS
- Isso porque o NDS é o repositório principal dos dados e porque o ETL do DDS é parametrizável
- É mais fácil de manter o master data em um repositório normalizado como o NDS e reconstruir o DDS pois lá existe pouca ou nenhuma redundância, e assim, atualiza-se apenas em um local

✘ Desvantagens

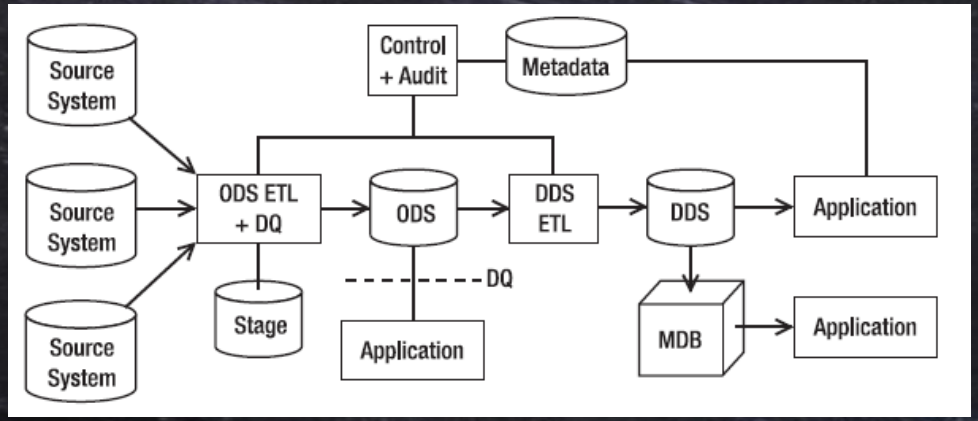
- A principal desvantagem é que se necessita maior esforço comparado a arquitetura DDS simples
- O esforço de construção de ETL praticamente dobra
- O esforço de modelagem é maior também

Arquitetura 2: NDS + DDS

- **Quando usar:**
  - **Usa-se NDS + DDS quando precisa-se construir diversos DDS para diferentes propósitos contendo um conjunto diferente de dados.**
  - **Também quando se necessita integrar dados de maneira normalizada e usar os dados integrados fora o DW dimensional**

Arquitetura 3: ODS + DDS

- Assim como NDS, ODS contém **tabelas de transação e tabelas master**
- Ao contrário do NDS que é interno, **ODS é híbrido, ou seja, acessível aos usuários finais**
- Eles podem ler dados do ODS, mas também podem atualizar esses dados
- Não deve-se atualizar dados vindos das fontes, mas sim dados gerados no processo do DW para complementar informações
- Nesta arquitetura, o **DDS é o repositório principal**, assim temos apenas um DDS



Arquitetura 3: ODS + DDS

- **Vantagens**
  - **A terceira forma normal é mais leve que no NDS pois contém apenas valores atuais do master data. Isto faz com que o desempenho tanto do ETL do ODS quanto do DDS seja melhor**
  - **O repositório relacional normalizado é atualizável por aplicações do usuário final, tornando possível suportar aplicações operacionais no nível de transação**

- **Desvantagens**
  - **A desvantagem desta arquitetura é que para criar um novo DDS menor, precisa-se obter a informação a partir do DDS principal e não pode utilizar os ETL existentes**

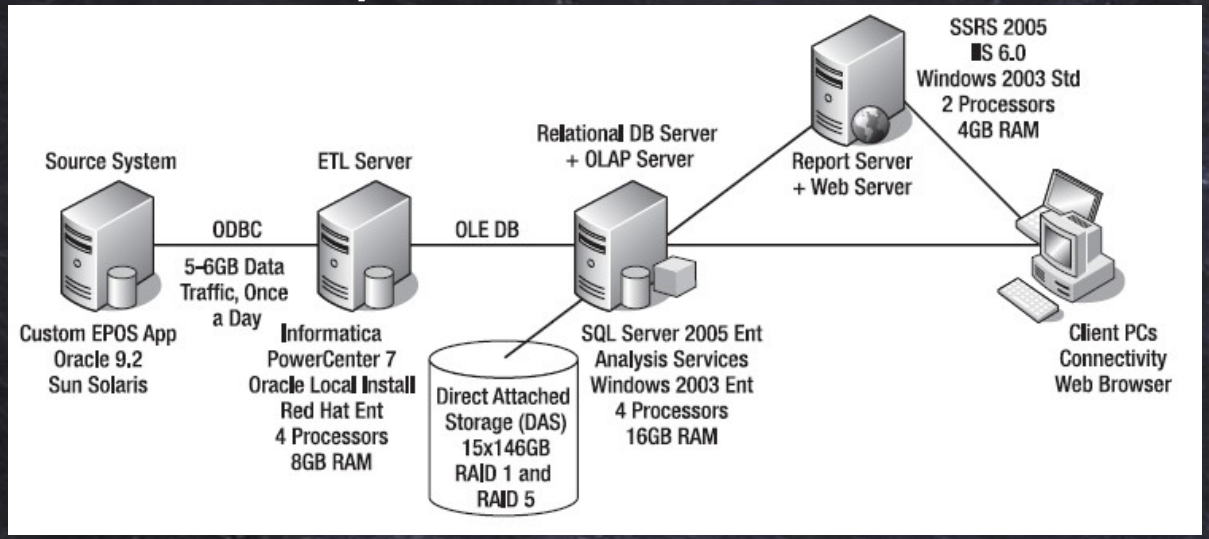
Arquitetura 3: ODS + DDS

- **Quando usar:**
  - **Usa-se esta arquitetura quando você precisa apenas um repositório dimensional e também precisa de um repositório centralizado e normalizado para ser usado com propósitos operacionais como um CRM por exemplo**

Arquitetura do sistema

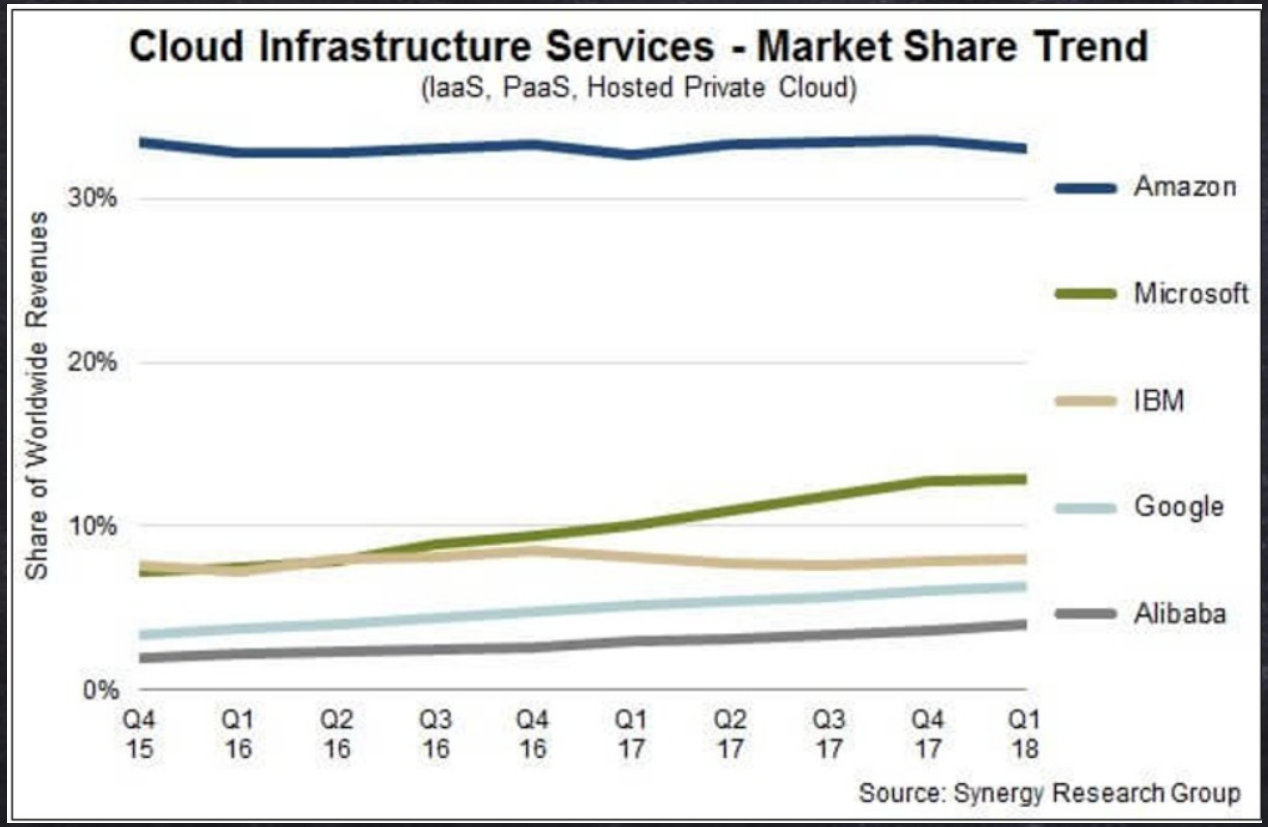
- **Depois de definida a arquitetura do fluxo de dados, deve-se projetar a arquitetura do sistema**
- **Isso é o arranjo físico e conexões entre servidores, rede, software, storage e clientes**
- **Projetar uma arquitetura de sistema requer conhecimento sobre hardware (servidores especificamente), redes (segurança e desempenho) e storage (SAN, RAID, backup)**

Arquitetura do sistema



- **Neste exemplo, a arquitetura do sistema é composta por três servidores: um para ETL, um para base de dados e um para relatórios.**
- **O sistema fonte é um ponto de vendas eletrônico**
- **Este exemplo mostra uma arquitetura típica de um sistema de tamanho médio**

Market Share – Cloud Computing



Atividade

- **A empresa “Dog plus plus” é uma franquia de sanduiches que conta com mais de 400 carrinhos de hot dog espalhados pela cidade de São Paulo. Cada venda feita é imediatamente reportada à um sistema integrado de Gestão que recepciona todas as informações de venda a partir de um app utilizado por cada carrinho. A alta cúpula da empresa observou, de forma rustica e grosseira, que alguns carrinhos vendem mais e outros vendem menos em diferentes dias da semana. Decidiu-se que deve ser projetado um DW para que apenas a Direção possa fazer consultas a partir de uma visão Dimensional. Você e sua equipe foram contratados para desenhar a Arquitetura de Fluxo de dados do DW. Desenhe o esboço de uma arquitetura de fluxo de dados (repositórios e processos/serviços), para atender a necessidade da empresa.**

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.
- DUTTA, P. DUTTA, P. Comparative Study of Cloud Services Offered by Amazon, Microsoft and Google. International Journal of Trend in Scientific Research and Development (IJTSRD), v. 3, Issue 3, April, 2019

Obrigado!

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