Roteiro da Apresentação

Boa tarde!

Meu nome é Carine e hoje eu venho apresentar meu trabalho de dissertação.

O título do trabalho é ETL4NoSQL: Um Framework para extração, transformação e carga de banco de dados NoSQL.

E a minha orientadora é a Professora Valéria.

A maioria dos SGBD de aceitação no mercado, segundo alguns autores, utilizam o modelo relacional, como o MySQL, o Oracle, SQL Server, entre outros.

Porém, os requisitos de dados para o desenvolvimento das ferramentas atuais vêm mudado significativamente, principalmente com o uso da Web. Como é o caso de aplicações como Twitter, Amazon, Facebook. Esses tipos de aplicações exigem grandes volumes de dados e respostas rápidas.

Com isso, surgiram novos segmentos de modelos de dados conhecidos por NoSQL.

Comparando os modelos relacionais com os modelos NoSQL temos vantagens e desvantagens de cada um. Nenhum autor arrisca dizer quem um modelo é melhor que o outro, apenas que cada um atendem as particularidades dos requisitos das aplicações.

Alguns autores afirmam que empresas coletam um vasto volume de dados, e que a análise desses dados é uma vantagem competitiva. Levando em consideração as mudanças dos requisitos das aplicações, como mencionado anteriormente, há uma necessidade de novas arquiteturas que deem suporte à decisão para essas aplicações.

E para oferecer suporte à decisão, uma alternativa bastante utilizada é a criação de Data Warehouse.

O Data Warehouse consiste em um depósito de dados utilizado para armazenar informações detalhadas de um determinado assunto, criando e organizando relatórios através de históricos auxiliando a tomada de decisão.

Os projetos de Data Warehouse consolidam dados de diferentes fontes de dados e cabem aos processos de ETL a extração, transformação e carga desses dados.

Os sistemas de ETL possuem a função de extrair dados de diversas fontes de dados, transformar esses dados conforme as regras de negócio determinado e por fim, carrega-los em um repositório de dados analíticos, que geralmente é Data Warehouse.

Segundo autores, os sistemas de ETL consomem cerca de 70% dos recursos para o desenvolvimento do DW. Porém, por muito tempo ele foi visto apenas como uma etapa para criação de DWs. Apenas a partir dos anos 2000 que a comunidade acadêmica passou a dar mais importância a ele. Existem várias ferramentas de ETL baseadas em GUI, tais como Clover, Talend, Pentaho, etc.

Então, quais as dificuldades encontradas ao lidar com essas ferramentas?

Essas ferramentas possuem limitações de modelagem. Elas possuem notações próprias que muitas vezes dificultam o projetista de ETL na hora de modelar transformações em domínios específicos.

A integração com os sistemas desenvolvidos pelas empresas também pode se tornar um desafio ao lidar com essas ferramentas de ETL.

Os sistemas de ETL frequentemente não dispõem de um ambiente integrado para modelagem dos processos e a extensão para as transformações dos dados.

Cada ferramenta possui sua interface gráfica e notação própria dificultando o aprendizado do uso de várias delas.

É comum que elas ofereçam apenas suporte aos SGBD relacionais, porém com a mudança dos requisitos das aplicações é importante dar suporte aos modelos NoSQL também.

Dessa forma, chegamos a motivação do nosso trabalho. De acordo com alguns autores, muitas empresas têm ignorado o uso de GUI, e estão optando pelo uso de uma linguagem de propósito geral para criar seus projetos de ETL.

Mas então quais são as dificuldades em adotar as linguagens de propósito geral e as soluções de ETL existentes?

Ao utilizar linguagem de programação de propósito geral, não há suporte aos fundamentos de ETL e seu reuso fica limitado ao escopo do projeto.

Já no caso das soluções de ETL existentes, elas oferecem pouco suporte, ou algumas, nenhum suporte aos bancos de dados NoSQL.

Além disso, as soluções de ETL muitas vezes apresentam dificuldade para customizar domínios de negócio específicos e a integração com outras aplicações, além de exigir conhecimento de notações diferentes para cada ambiente.

Com isso, definimos os objetivos da nossa pesquisa. Como objetivo geral temos:

Especificar um framework programável, flexível e integrado para modela e execução dos processos de ETL para bancos de dados NoSQL.

Nossos objetivos específicos são: Apresentar um estudo experimental de software, a fim de caracterizar as principais funcionalidades das ferramentas de ETL para BDs NoSQL.

E prover dois ambientes de ETL a partir do framework proposto para facilitar a extração, transformação e carga de dados em DW modelados pelo esquema estrela.

Quadro comparativo entre as ferramentas de ETL – revisão de literatura.

Para o nosso trabalho propomos elencar os requisitos de software do ETL4NoSQL, as modelagens do domínio e especificação, a especificação dos componentes, interfaces de programação, o estudo experimental de software e as especializações a partir do ETL4NoSQL. Estarei explicando esses itens com mais detalhe a seguir.

Os requisitos de software elencados para o ETL4NoSQL foram:

A entidade Modelagem faz a leitura de uma ou mais Fontes de dados, executa nenhuma ou muitas Operações. As Operações por sua vez são processadas por um ou mais Processamentos de forma a serem Processamentos Centralizados ou processamentos Distribuídos. E por fim, a Modelagem carrega o resultado das Operações processadas à um ou mais Destinos.

Os casos de uso expressam as funcionalidades fundamentais ao desenvolvimento e uso do ETL4NoSQL. Eles permitem ao programador a visão do que é imprescindível ao implementar e determinar as interfaces de sistema e operações do framework. Um conjunto de casos de uso foram identificados tais como: Ler fonte de dados, Escrever no destino, Modelar dados, Executar operação e Processar operações.

Ao construir o modelo comportamental é possível identificar os conceitos com comportamentos mais relevantes para o negócio, bem como os estados e eventos que disparam as transições entre os estados (SOUZA GIMENES; HUZITA (2005)). Dessa forma, apresentamos o diagrama de estados do ETL4NoSQL, nele podemos ver as transições de leitura da fonte de dados, validação e identificação dos dados, assim como o tratamento caso os dados não possam ser identificados. Subsequente a isso, podemos ver as transições do armazenamento dos dados para o processamento, a criação dos processos de ETL, a escolha da forma de processamento, execução das operações, e também o tratamento para as operações que não puderem ser executadas. Finalmente, a transição da carga dos dados pode ser feita na base de destino seguido da mensagem de tratamento, caso haja sucesso ou não na execução.

A modelagem de especificação visa definir, em um nível alto de abstração, os serviços oferecidos pelos componentes. Dessa forma, é possível determinar a arquitetura e especificar os seus componentes. É importante dar ênfase a especificação das interfaces, pois isso contribui para uma clara separação entre os componentes, e também, para assegurar o princípio de encapsulamento de dados e comportamento. CHEESMAN; DANIELS (2001) divide a modelagem da especificação em três estágios: Identificação de componentes, Interfaces de Sistemas e Interfaces de Negócio.

As interfaces de sistemas são identificadas por meio dos casos de uso, mencionados na fase de licitação de requisitos. Assim, identificamos as interfaces:

As interfaces de negócio são identificadas por meio do modelo conceitual, cada entidade identificada no modelo conceitual é uma interface de negócio.

Assim, após definir as interfaces de sistemas e de negócio podemos determinar a arquitetura dos componentes.

Depois de definir as operações foi possível criar o diagrama de especificação das interfaces. Essas especificações referem-se ao uso dos componentes. Elas representam o que o usuário precisa saber sobre os componentes. Aqui nós temos as interfaces de negócio identificadas, as interações entre os componentes e as operações.