Explicar o processo de migração de dados, explicando de forma detalhada as suas principais etapas - extração, transformação e carregamento

Na perspetiva do grupo, esta migração tem como objetivo a consulta dos dados, através da eficiência da base NoSQL e também a eliminação dos JOINs existentes na base de dados relacional implementada anteriormente. Numa fase inicial de avaliação deste processo, identificamos duas entidades principais sobre as quais realizamos as várias queries, Cliente e Viagem. Assim, definimos a existência de duas Collections. Dentro de cada collection tentamos agregar toda a informação possível para que não fosse necessário recorrer a “JOINs” entre collections.

Feita a avaliação dos requisitos desta migração procedeu-se à execução da mesma. Para isso foi realizado um script na linguagem JAVA (não a mais apropriada), utilizando bibliotecas e drivers existentes relativas ao motor MySQL (extração), classes auxiliares para facilitar o manuseamento dos dados (transformação) e por fim bibliotecas relativas ao motor MongoDB (carregamento).

Para uma melhor explicação do processo, apresenta-se em seguida ……… de uma row de cada uma das collections referidas anteriormente.

IMAGENS

Em relação à Collection Cliente, e mais especificamente a cada row, é guardado o id do cliente (que serve também como identificador para o objeto mongo), o seu nome, email, nif, password e uma lista de todos os bilhetes adquiridos por si. Em cada bilhete presente nesta lista está presente o seu id, preço, data de aquisição, classe, número, data de partida, data de chegada, duração, origem e destino. Para a construção de cada uma das rows são então montados os objetos auxiliares e introduzidos individualmente, pelo que não existem problemas de falta de memória relativos a este processo.

Para extração dos dados o script foi o seguinte:

IMAGEM

A primeira query presente na figura serve, como é obvio, para selecionar todos os clientes registados na base de dados. O ‘ResultSet rs’ representa tal resultado e iterando sobre ele podemos ir construindo a informação relativa a cada Cliente. A segunda query é utilizada para construir a lista dos bilhetes associados a um cliente. Esta query vai buscar informação de cada um dos bilhetes associados ao cliente em causa assim como informação relativa à viagem a que o bilhete pertence para contemplar toda a informação necessária à execução correta das queries sobre esta coleção sem ter necessidade de recorrer a outra. Construída a informação relativa a um cliente é então necessária a sua introdução na nova base de dados. Este processo é feito através do método ‘mongoAddCliente(Cliente c)’ que é explicado em seguida e cuja implementação pode ser vista na figura em seguida.



Inicialmente é necessário obter ligação com a base de dados Mongo e posteriormente obter a coleção relativa aos clientes. Para introduzir dados é necessário primeiro convertê-los e isso é feito introduzindo-os em um “BasicBDObject”. O processo é terminado pela inserção do objeto criado, ou seja, a nova row a ser introduzida através do método “insert(doc)” que é passado à coleção anteriormento obtida.

Em relação à Collection Viagem, e novamente relativo a cada row, é guardado o id da viagem, o seu preço base, datas de partida e chegada, duração, o comboio pela qual é realizada, origem e destino, e uma lista de bilhetes. Esta lista, ao contrário da lista presente em cada row da Collection Cliente, não representa todos os bilhetes comprados para a viagem em causa, mas sim uma lista de todos os possíveis bilhetes. O que diferencia os bilhetes comprados dos não comprados é ……… O processo de construção de cada row é semelhante ao explicado no caso de Cliente.

IMAGEM

SOBRAL EXPLICA AHAH