ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO

Licenciatura em Engenharia Informática

Processamento Estruturado de Informação

Trabalho Prático 2

Grupo: 27

João Lopes 8190221

Nuno Silva 8190285

Tiago Leite 8190338

Conteúdo

1.Contextualização

O objetivo deste trabalho é modelar a informação de uma base dados relacionadas com vendas de produtos, para uma base de dados orientadas a documentos (Mongo DB) de modo a facilitar o máximo possível pesquisas de informação e atualização de dados. O caso de estudo é a gestão e tratamento de informação de uma loja de vendas de produtos. Dados relacionados com clientes, produtos, detalhes de vendas vão ser utilizados para o apoio e tomadas de decisão relacionados com os produtos comercializados.

2. Abordagem da estruturação da base de dados

2.1 Identificação das entidades e os seus relacionamentos

As principais entidades na base de dados são os clientes, os produtos e as vendas. Os dados serão armazenados no MongoDB, uma base de dados orientada por documentos, por isso o objetivo será embutir sempre que existir um relacionamento entre as entidades. No entanto, poderão haver algumas situações em que fará sentido referenciar, de modo a tentar equilibrar as necessidades do utilizador e o desempenho da base de dados.

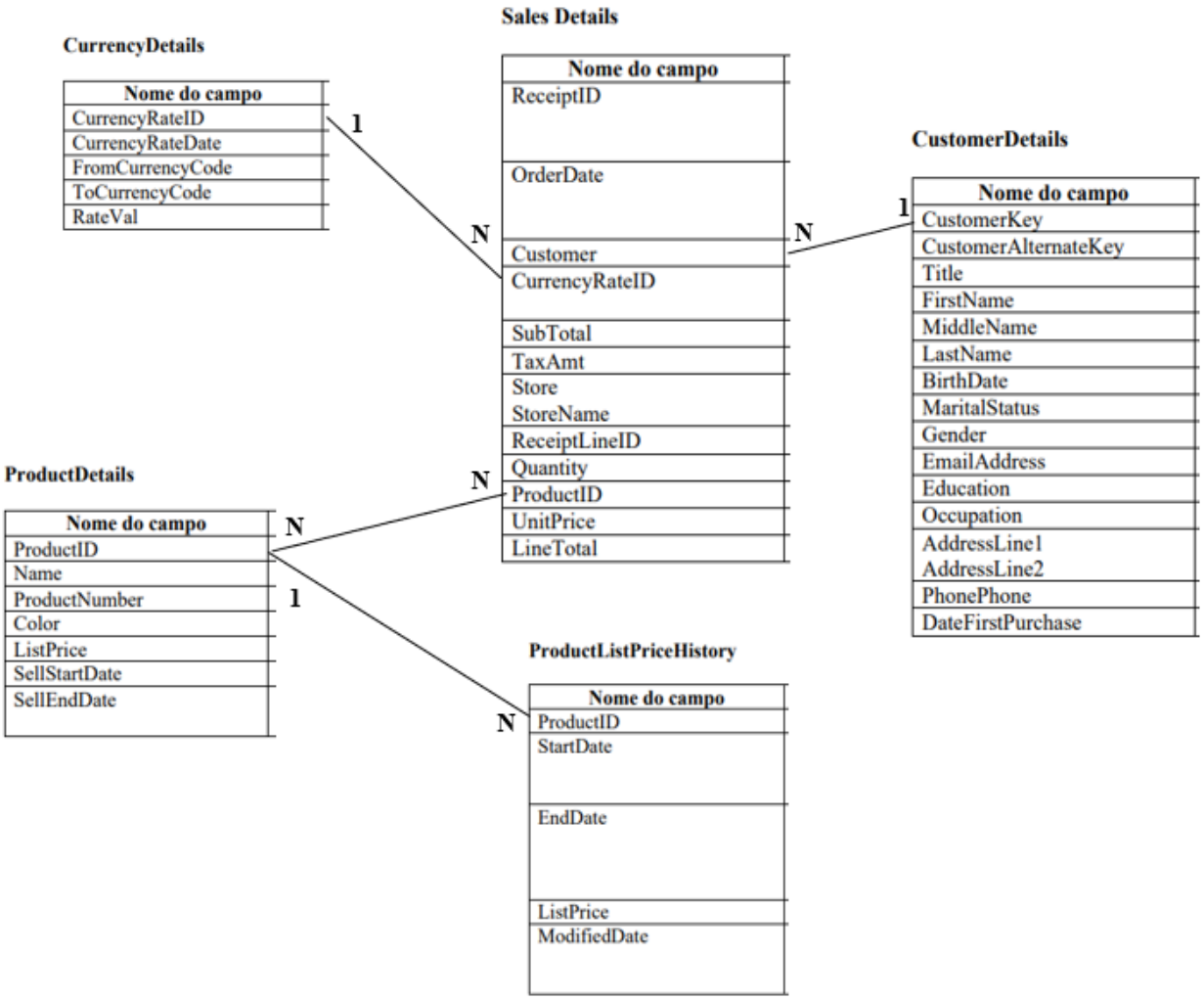

Figure 1 - Relacionamentos entre documentos

3. Identificação de requisitos

Considerando que se trata de uma loja de vendas de produtos, faz sentido agregar alguma parte da informação de outras entidades no documento de detalhes das vendas de modo a haver um relacionamento entre cliente e o produto. Estando toda a informação no mesmo documento fica facilitada a tarefa da procura de dados entre clientes e produtos em simultâneo.

Visto que se trata de uma loja de venda de produtos, esta entidade é o foco do programa e a taxa de pesquisa provavelmente será muito alta, por isso embutimos alguma informação no documento de vendas como o nome e o id, para serem efetuadas pesquisas relacionadas com as vendas em que estes estão envolvidos. Útil por exemplo para o utilizador perceber qual o produto que é mais comprado, num determinado período, sem a necessidade de efetuar mais do que uma pesquisa. No entanto, também é muito provável que neste tipo de negócios de vendas de produtos, haja necessidade de atualização frequente de alguns campos como o preço do produto. Por isso optamos por não embutir este campo porque tornaria este processo de atualização muito demorado, mantendo toda a coleção de produtos para facilitar também as consultas por produtos.

O histórico de preços da lista de produtos será integrado no documento dos produtos visto que se trata de informação do mesmo tipo, podendo assim ser consultada mais facilmente a informação do histórico de valores de cada produto. ---

Relativamente aos clientes, a taxa de atualização da informação é muito baixa e além disso é muito provável consultas a envolverem os clientes e as suas compras, por isso faz sentido embutir no documento das vendas. Contudo, para o documento das vendas não ficar muito extenso decidimos selecionar apenas a informação mais importante e que é mais frequentemente mais consultada.

4. Modelação dos dados:

Após os cenários identificados anteriormente, o documento relativo a "CurrencyDetails" irá ser embutido no documento de "SalesDetails"(fig.2) visto que uma venda apenas tem relacionada um único documento de detalhes de câmbio (relação de 1-N), e este documento é relativamente pequeno. Assim é possível apenas numa consulta fazer a leitura da venda e dos seus detalhes de cambio. Apesar de esta abordagem implicar a repetição de informação em vários documentos de vendas, optamos por embutir porque não são esperadas grandes atualizações de dados nestes campos.

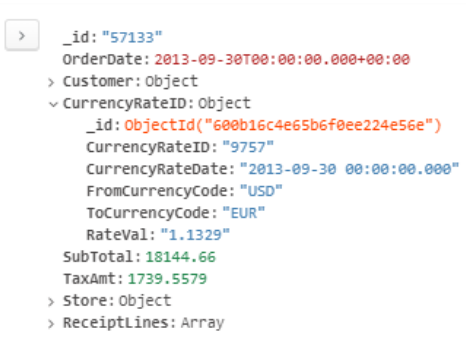

Figure 2 - Sales Details Document

O documento de "ProductListPriceHistory" também irá ser embutido no documento de "ProductDetails"(fig.3). O relacionamento será de 1-N, ou seja, um produto poderá ter vários documentos de atualização do preço, no entanto, um documento de atualização do preço apenas tem associado um produto. Assim toda a informação de um produto fica armazenada num único documento, facilitando assim a consulta de informação de um produto. Para além disso, o documento de "ProductListPriceHistory" não terá muitas atualizações de informação, devido a este fazer apenas o registo de todas as alterações do preço, não sendo previsível que haja muitas modificações nos campos.

Figure 3 - Products Details Document

O documento de vendas poderá ter vários produtos e cada produto poderá estar associado a várias vendas, por isso, estamos perante um relacionamento de N-N. Ao contrário dos casos anteriores, o documento "Sales Details" vai guardar apenas o nome e o número de produto num array. Optamos por esta técnica porque, considerando que é uma loja de vendas de produtos, é previsível que haja muitas atualizações nos produtos a serem vendidos, o que a técnica de embutir tornaria este processo bastante demorado. No entanto vamos embutir o nome e o código do produto para auxiliar nas pesquisas em simultâneo entre vendas e produtos.

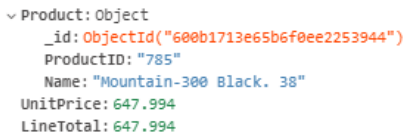

Figure 4 - Products embedded

O documento "CustomerDetails" será embutido no documento de "SalesDetails", visto que uma venda apenas tem um cliente associado (relação de 1-N), no entanto, optamos por dividir em dois documentos, porque a informação do cliente ainda é bastante extensa. Por isso será colocado nas vendas apenas os campos que normalmente são mais consultados e noutro documento as informações do cliente que normalmente são menos frequentes serem consultadas - padrão Subset. Assim evitamos que o documento vendas fique demasiado extenso e com informação que provavelmente não será muito pesquisada. Para além disso, assim é possível poupar espaço no documento vendas e ainda assim ser possível efetuar consultas de cliente e vendas em simultâneo.

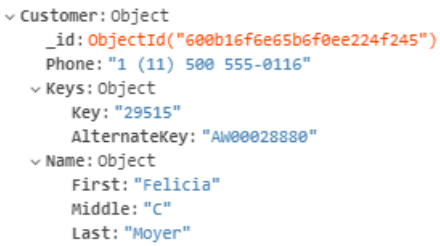

Figure 5 - Customer embedded

5. Processo de transformação

Alguns campos que partilham características em comum foram agrupados e alguns tipos de variáveis foram corrigidos para o tipo de dados que armazenavam.

A transformação efetuada para a estruturação dos dados foi pela seguinte ordem apresentada. A numeração da transformação apresentada no relatório corresponde à numeração presente no ficheiro "SalesTP.js".

Ficheiro: TPSales.js

Transformação 1: - Embutir apenas os campos name e ProductID da coleção ProductsDetails na coleção Sales Details

1. Lookup para juntar a coleção ProductDetails na coleção Sales Details.
2. Unset para eliminar os campos que não irão ser embutidos como ProductNumber, Product.Color, ListPrice, SellStartDate, SellEndDate.
3. Unwind para retirar cada produto do array criado pelo Lookup.
4. Group para agrupar todas as linhas de receita com o mesmo ReceiptID num único documento e Push para acumular todas as ReceiptLines com o mesmo ReceiptID.

Nota: De seguida o resultado anterior é guardado numa nova coleção e a coleção antiga é apagada.

Transformação 2: - Agrupar atributos comuns do cliente

1.Project do id, Title, Keys com os campos Key e AlternateKey agrupados, Title, Name com os campos First, Middle, Last agrupados, Birthdate, MaritalStatus, Gender, EmailAddress, Education, Ocupation, Address com os campos Line1 e Line2 agrupados, Phone e DateFirstPurchase.

Nota: De seguida o resultado anterior é guardado numa nova coleção e a coleção antiga é apagada.

Transformação 3 – Embutir alguns campos da coleção Costumer em SalesDetails.

1. Lookup para juntar a coleção Customer na coleção SalesDetails.
2. Unwind de Customer para retirar o objeto Customer do array criado pelo Lookup.
3.Unset para eliminar os campos Title, MaritalStatus, Gender, Education, BirthDate, Address, DateFirst, DateFirstPurchase, EmailAddress.

Nota: De seguida o resultado anterior é guardado numa nova coleção e a coleção antiga é apagada.

Transformação 4 – Embutir coleção ProductPriceHistory em ProductDetails.

1. Lookup para embutir a coleção ProductPriceHistory em ProductDetails

Nota: De seguida o resultado anterior é guardado numa nova coleção e a coleção antiga é apagada. Depois são inseridos os produtos numa coleção nova que não têm PriceHistory associado.

Transformação 5 – Agrupar os produtos que têm um PriceHistory associado.

1.Unwind para retirar o objeto do array PriceHistory da coleção Products.
2.Group para agrupar os produtos que têm o mesmo ProductID e push para acumular todos os campos PriceHistory com o mesmo ProductID.

Nota: De seguida o resultado anterior é guardado na coleção anterior juntamente com os produtos que não têm um PriceHistory associado. Primeiro optou-se por adicionar os produtos sem PriceHistory associado porque o comando unwind do array PriceHistory elimina todos os documento que tem valor NULL. Depois a coleção antiga é apagada.

Transformação 6 – Embutir o CurrencyDetails no SalesDetails.

1.Lookup da coleção CurrencyDetails na coleção SalesDetails.

Nota: De seguida o resultado anterior é guardado numa nova coleção e a coleção antiga é apagada. Depois são inseridos os produtos numa coleção nova que não têm PriceHistory associado.

São agregados numa variável o cursor com os documentos SalesDetails sem CurrencyRateID associado.

Transformação 7 – Retirar objeto do array CurrencyDetails.

1.Unwind da coleção CurrencyDetails na coleção SalesDetails, e de seguida é adicionado o resultado a uma nova coleção.

O cursor das vendas sem CurrencyRateID que ficaram armazenadas na variável são adicionadas à mesma coleção. Foi dotada a mesma estratégia de adicionar primeiro as vendas sem CurrencyRateID associado visto o comando unwind elimina todos os documentos com um array NULL.

Transformação 8 – Alterar o preço do produto que tenha valor de 0 pelo mais recente no histórico de preços diferente de 0.

1.Match dos produtos iguais a zero e histórico de preço diferente de 0.
2.Set do valor ListPrice pelo último valor do array do histórico de preço
3.Unwind do ListPrice para retirar o array assocaido ao campo.

Nota: De seguida o resultado anterior é guardado numa nova coleção e a coleção antiga é apagada.

Transformação 9 – Alterar o preço do produto que tenha valor de 0 pelo mais recente no histórico de preços diferente de 0.

1.Match dos preços com valores diferentes de zero.

Nota: De seguida o resultado anterior é guardado numa nova coleção e a coleção antiga é apagada.

Transformação 10 – Colocar o tipo de dados correto do campo ListPrice

1.addFields do campo ListPrice e toDouble da String ListPrice.

Nota: De seguida o resultado anterior é guardado numa nova coleção e a coleção antiga é apagada.

Todas as novas coleções geradas entre as transformações foram criadas com o recurso forEach.

```
newProducts.forEach(function (doc) {
    db.ProductsList.insertOne(doc);
});
```

Figure 6 - Cursor for creation a new collection

6. Consultas de Informação

Ficheiro: Consultas.js

Transformação 1 – Obter uma listagem do número total de unidade vendidas por produto para um determinado período. Os resultados devem estar ordenados de forma descendente pelo número de unidades vendidas.

1. Match de uma data de encomenda entre 2012-02-10 e 2013-02-10.
2. Unwind de ReceiptLines para retirar cada objeto desse array para um novo documento.
3. Group dos produtos com o mesmo ID e Name. Count para contabilizar o número total de unidades vendidas desse produto.
4. Sort para ordenar por ordem descendente o total de produtos.

Transformação 2 - Obter uma listagem do número total de unidade vendidas por cliente para um determinado período. Os resultados devem estar ordenados de forma descendente pelo número de unidades vendidas.

1. Match de uma data de encomenda entre 2012-02-10 e 2013-02-10.
2. Unwind de ReceiptLines para retirar cada objeto desse array para um novo documento.
3. Group do cliente com o mesmo ID e Name. Count para contabilizar o número total de unidades vendidas desse produto.
4. Sort para ordenar por ordem descendente o total de produtos.

Transformação 3 - Obter uma listagem das vendas em que não existe um cliente válido associado.

1. Lookup para juntar a coleção Customer na coleção Sales. Se não existir um cliente associado na venda, o Lookup coloca o campo gerado com um array vazio.
2. Match do campo gerado e filtrámos os array que não tem nenhum cliente.
3. Project dos dados da venda que não tem cliente associado.

Transformação 4 - Obter uma listagem dos produtos descontinuados.

1. Find do campo SellEndDate da coleção ProductsList, que tem uma data de fim.

Transformação 5 - Obter uma listagem dos clientes que não compram nenhum produto há mais de um mês.

1. Group do cliente com o mesmo ID na coleção Sales, e guardar a data mais recente de uma compra.
2. Match do campo gerado com, a data mais recente e filtrar aquelas que são menores que a data de há um mês.
3. Project do cliente e da data da última encomenda.

Transformação 6 - Obter uma listagem dos produtos que não são vendidos há mais de uma semana.

1. Unwind para retornar num documento cada ReceiptLine da coleção Sales.
2. Group do produto com o mesmo ID na coleção Sales, e guardar a data mais recente de uma compra.
3. Match do campo gerado com, a data mais recente e filtrar aquelas que são menores que a data de há um mês.
4. Project do produto e da data da última encomenda.

7. Mongo Atlas – Charts

Link Dashboard:
https://charts.mongodb.com/charts-pei_tp2-oprmk/public/dashboards/bb1a221a-fe6d-4af2-b100-a9e21674a77f

6. Discussão de resultados

A abordagem adotada para a estruturação da informação tens vantagens e também tem as suas desvantagens. Ao embutir o ListPriceHistory existe sempre a desvantagem de o número de documentos estar sempre a aumentar pois em cada atualização de informação podem ser adicionados mais alterações do preço do produto. Ainda assim optou-se por embutir porque não é comum serem feitas um número tão elevado de alterações de modo a atingir o limite máximo de um documento. Foram embutidos também o cliente e o nome dos produtos no documento das vendas, e apesar de haver repetição da informação, a pesquisa da informação das varias entidades - Vendas, produtos e cliente, podem ser consultadas com apenas uma pesquisa. Uma desvantagem seria se fosse necessário atualização do nome de um produto, pois seria necessário percorrer todos os documentos das vendas.

7. Conclusão

Concluído o trabalho, foram sentidas algumas dificuldades com a idealização da melhor estrutura de dados para o problema apresentado, visto existirem prós e contras para qualquer das abordagens a seguir. Também na ferramenta do mongo Atlas foram encontradas algumas dificuldades na construção de alguns gráficos pedidos, devido também à falta de tempo.