P.PORTO

ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO

Relatório

Processamento Estruturado de Informação

2º Ano – 1º Semestre

Afonso Santos 8150025 - LSIRC

Nuno Josefino 8150189 - LEI

Hugo Silva 8130142 - LEI

Índice

Introdução

Nesta segunda entrega, podemos observar algumas alterações nos documentos .xsd que por sua vez vão mudar ligeiramente a estrutura dos ficheiros.

Temos também o desenvolvimento inicial das querys de pesquisa que vão ser necessárias para este projeto

UserDetails

Não foram necessárias alterações neste ficheiro para a segunda parte do projeto

CurrencyDetails

Não foram necessárias alterações neste ficheiro para a segunda parte do projeto

ProductDetails

Não foram necessárias alterações neste ficheiro para a segunda parte do projeto

SalesDetails

Foi alterado o documento salesDetails.xsd de modo a suportar os novos requisitos

O valor “order” foi substituído pelo valor “quantity” que apresenta as mesmas características e validações

```
<xs:element name="quantity">
    <xs:simpleType>
        <xs:restriction base="xs:integer">
            <xs:minInclusive value="0"/>
            <xs:maxInclusive value="100"/>
        </xs:restriction>
    </xs:simpleType>
```

StoreDetails

Foi alterado o documento storeDetails.xsd de modo a suportar os novos requisitos

Foram adicionados os valores “totalProductSold”, “totalSaleValue” e “averageSaleValue” que vão servir para representar valores de vendas das lojas

```
<xs:element name="totalProductSold">
    <xs:simpleType>
        <xs:restriction base="xs:integer">
            <xs:maxLength value="250"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="totalSaleValue">
    <xs:simpleType>
        <xs:restriction base="xs:decimal">
            <xs:minInclusive value="0"/>
            <xs:maxInclusive value="unbounded"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="averageSaleValue">
    <xs:simpleType>
        <xs:restriction base="xs:decimal">
            <xs:minInclusive value="0"/>
            <xs:maxInclusive value="unbounded"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
```

productListPriceHistory

Foi adicionado o ficheiro productListPriceHistory.xsd que trata e valida a informação do documento ProductListPriceHistory.csv

Uma vez que este ficheiro utiliza informações utilizadas e definidas em ficheiros anteriores, recorremos á ultização de imports para os reutilizar e a utilização de namespaces para que não haja conflito entre os elementos.

Utilizamos o “:p” para caracterizar elementos do schema productDetails.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://www.estg.ipp.pt/prodListPriceHistory"
targetNamespace="http://www.estg.ipp.pt/prodListPriceHistory"
xmlns:p="http://www.estg.ipp.pt/productDetails" elementFormDefault="qualified">
    <xs:import schemaLocation="productDetails.xsd" namespace="http://www.estg.ipp.pt/productDetails" />
    <xs:element name="listPrice">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="productID" type="p:prodID" />
                <xs:element name="startDate" type="p:sellStartDate" />
                <xs:element name="endDate" type="p:sellEndDate" />
                <xs:element name="lPrice" type="p:listPrice" />
                <xs:element name="modifiedDate">
                    <xs:simpleType>
                        <xs:restriction base="xs:dateTime">
                            <xs:minInclusive value="1900-01-01T00 : 00 : 00.0Z" />
                            <xs:maxInclusive value="2200-01-01T00 : 00 : 00.0Z" />
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

CRIAÇÃO DA BASE DE DADOS E COLLECTIONS

use db BikeOnTrack \\cria a base de dados BikeOnTrack no MongoDB

db.createCollection("CurrencyDetails") \\cria a colleção CurrencyDetails(mongoDB shell)

db.createCollection("ProductDetails") \\cria a colleção ProductDetails(mongoDB shell)

db.createCollection("ProductListPriceHistory") \\cria a colleção ProductListPriceHistory(mongoDB shell)

db.createCollection("SalesDetails") \\cria a colleção SalesDetails(mongoDB shell)

IMPORTAÇÃO DOS DADOS CSV PARA AS RESPETIVAS COLEÇÕES

mongoimport -d BikeOnTrack -c CurrencyDetails ---type csv --file ~/Desktop/moodle/CurrencyDetails.csv --headerline

\\importa os dados csv do ficheiro CurrencyDetails para a coleção CurrencyDetails da base de dados BikeOnTrack(cmd shell)

mongoimport -d BikeOnTrack -c ProductDetails ---type csv --file ~/Desktop/moodle/ProductDetails.csv --headerline

\\importa os dados csv do ficheiro ProductDetails para a coleção ProductDetails da base de dados BikeOnTrack(cmd shell)

mongoimport -d BikeOnTrack -c ProductListPriceHistory --type csv --file ~/Desktop/moodle/ProductListPriceHistory.csv --headerline

\\importa os dados csv do ficheiro ProductListPriceHistory para a coleção ProductListPriceHistory da base de dados BikeOnTrack(cmd shell)

mongoimport -d BikeOnTrack -c SalesDetails ---type csv --file ~/Desktop/moodle/SalesDetails.csv --headerline

\\importa os dados csv do ficheiro SalesDetails para a coleção SalesDetails da base de dados BikeOnTrack(cmd shell)

Querys de pesquisa

```
db.SalesDetails.aggregate(

      { $match: { StoreName: "Better Bike Shop" } },
        { $match: { OrderDate: { $regex: /2011-08/ }}},
        {$facet: {
            "Total de produtos":[
                {$group : {"_id":"$ReceiptID", "total":{$sum: "$Quantity"}}
                }
            ],
            "Numero de produtos":[
               { $count: "ReceiptLineID" }
            ] ,
            "Média das vendas":[
                {$group : {"_id":"$ReceiptID", "total":{$avg: "$LineTotal"}}
                }
            ],
            "Total de clientes":[
                {$group : {"_id":"$Costumer"}
                },
                {$count: "Costumer"
                }
            ],
        "Valor vendido por cliente":[
            {$group:{"_id":"$Costumer", "total":{$sum: "$LineTotal"}}
        },
            {$sort:{total:-1}
        }
        ],
        "Lista dos produtos":[
            {$group:{"_id":"$ProductID", "total":{$sum: "$Quantity"}}
        },
            {$sort:{total:- 1}
        }
        ],
        "Lista de vendas":[
            {$match:{}
            }
        ]
        }
        }
).pretty()
```

db.SalesDetails.aggregate([{$match:{ StoreName: "Better Bike Shop"}},

{$match:{OrderDate:{$regex:/2011-08/}}}]).pretty()

Mostra os dados do SalesDetails para uma loja("Better Bike Shop" neste exemplo) num certo mês (2011-08 neste exemplo)

db.SalesDetails.aggregate([{$match:{StoreName:"Better Bike Shop"}},

{$match:{OrderDate:{$regex:/2011-08/}}},

{$group:{"_id":"$ReceiptID", "total":{$sum: "$Quantity"}}}])

Para uma loja especifica("Better Bike Shop" neste exemplo), num mes especifico(2011-08 neste exemplo) mostra a quantidade total de produtos vendidos

{$facet: {"Total de produtos":[{$group:{"_id":"$ReceiptID", "total":{$sum: "$Quantity"}}}],

Mostra o total de produtos vendidos

"Numero de produtos":[{$count: "ReceiptLineID" }],

Mostra o numero de produtos diferentes vendidos

"Média das vendas":[{$group:{"_id":"$ReceiptID", "total":{$avg: "$LineTotal"}}}],

Mostra a média das vendas

"Lista de vendas":[{$match:{}}]}}).pretty()

Mostra todas as vendas feitas

"Total de clientes":[{$group : {"_id":"$Costumer"}},{$count: "Costumer"}],

Mostra o total de clientes a quem a loja vendeu produtos

"Valor vendido por cliente":[{$group:{"_id":"$Costumer", "total":{$sum: "$LineTotal"}}},

{$sort:{total:-1}}],

Mostra o valor vendido a cada cliente por ordem decrescente

"Lista dos produtos":[{$group:{"_id":"$ProductID", "total":{$sum: "$Quantity"}}},

{$sort:{total:- 1}}],

Mostra a lista com a quantidade de produtos vendidos por ordem decrescente

"Lista de vendas":[{$match:{} }]

Mostra a lista com todas as informações das vendas para aquela loja naquele mês especifico

Para obter os resultados em relação a todas as lojas, apenas temos que remover o match que especifica a loja.

Conclusão

Para a próxima entrega, a entrega final, vai ser elaborado um método que vai converter os resultados dos querys para ficheiros XML tendo por base as regras definidas nos ficheiros .xsd