

|  |
| --- |
| Complementos de Bases de Dados 2022/2023  Licenciatura em Engª. Informática |
| 1ª Fase Relatório Técnico |

Turma: 05

Horário de Laboratório: 4ºFeira 16:30

Docente: Gabriel Pestana

Grupo

Nº201901111, António Rebelo  
Nº201901953, André Pauli

Nº202101294, Uliana Pyrohovska

# Introdução

*Explicação do projeto em alto nível, mencionando o âmbito, algumas funções e os objetivos gerais, de acordo com as assunções ou interpretações que formam tomadas na interpretação do enunciado do projeto.*

A base de dados original foi-nos dada num estado de pouca manutenção e com vários erros de normalização, por isso como âmbito para o projeto, criamos uma base de dados que segue as regras da normalização. Com esta nova base de dados, novos utilizadores poderão utilizar os dados antigos sem depararem-se com problemas a cerca da base de dados em si.

Para alcançar este objetivo, para além de normalizar a base de dados antiga, também introduzimos novas funçõespara facilitar o uso da base de dados. Por exemplo, introduzimos formas para adicionar novas vendas e formas automáticas que verificam possíveis erros dentro das novas vendas introduzidas.

# Especificação de Requisitos

*Especificar os requisitos funcionais apresentados no enunciado do projeto. Estes requisitos devem incluir restrições de integridade ou regras de validação de informação/processos de negócio. Os requisitos que são propostos como melhoria aos expostos no enunciado devem ser incluídos e identificados por RM##.*

|  |  |  |
| --- | --- | --- |
| R01 | O sistema deverá permitir a autenticação de um utilizador no sistema com base do seu email e password | *S* |
| R02 | O sistema deverá armazenar a password seguramente mediante de uma função de encriptação | S |
| R03 | O sistema deverá permitir a adição de novos utilizadores com base no ID do cliente, uma password e um email | S |
| R04 | O sistema deverá permitir a remoção de utilizadores existentes a partir do seu ID de utilizador | *S* |
| R05 | O sistema deverá permitir a edição de utilizadores com base no seu ID de utilizador, o novo email, nova password e novo customer ID | *S* |
| R06 | O sistema deverá implementar uma nova funcionalidade promoção que irá descrever valores de promoção, data de inicio e fim | *S* |
| R07 | O sistema deverá permitir alterar as datas de inicio e fim de uma promoção ativa | *S* |
| R08 | O sistema deve expor ferramentar para criar um novo cabeçalho de venda (SaleHeader) | *S* |
| R09 | O sistema deve expor ferramentas para adicionar novas vendas a um cabeçalho de venda (criar SaleDetails’) | *S* |
| R10 | O sistema deve poder alterar a quantia de um produto numa venda pelo id de venda | *S* |
| R11 | O sistema deve poder remover um produto numa venda (SaleDetails) e ainda ter a opção de remover logo o cabeçalho todo caso seja o único produto na venda | *S* |
| R12 | O sistema deve calcular e expor o preço total de uma venda | *S* |
| R13 | O sistema deve seguir a regra de negócio em que a data de entrega de um produto deve ser menor ou igual à data prevista | *S* |
| R14 | O sistema não deve permitir a venda combinada de produtos “Chiller Stock” e “Non-Chiller stock” | *S* |
| R15 | O sistema deverá efetuar o tratamento de erros nas diferentes funções e procedimentos, guardar o erro e expô-lo ao utilizador com uma mensagem personalizada e um timeStamp | *S* |
| R16 | O sistema deve apenas contabilizar uma venda, mesmo que esta contenha vários produtos associados | *S* |
| R17 | O sistema deve fornecer ferramentas geradoras para as operações de inserção, atualização e remoção de registos nas diferentes tabelas. A função geradora deverá receber como parâmetro uma tabela e uma operação e o seu output deverá ser o código a ser corrido | *S* |
| *R18* | O sistema deverá expor estatísticas de uma determinada venda (SaleHeader) | *S* |

# Alterações / Melhorias ao Relatório da 1ªa Fase

Tabela de Utilizadores criada com intuito de tornar a distribuição de utilizadores mais fácil. Agora cada cliente e 0empregado herda o id da *UserTable* onde o seu nome preferido é guardado. Devido a esta alteração, os comandos da migração relevantes a estes dados foram alterados.

*Views* das sales foram melhoradas ao adicionar uma nova *view “ItemsDiscounted”* que mostra as sales cujos items estão com disconto de promoção.

Foram implementadas *schemas* e foi adicionado descrições ao relatório. O diagrama, foi melhorado e os ficheiros foram numerados por ordem de que são usados.

# Modelo Relacional (*Modelo de dados*)

## Diagrama do Modelo Entidade Relação

*Imagem do diagrama entidade relação (opcional).*

## Diagrama do Modelo Relacional

# Definição do Layout

*Discriminando a projeção de necessidades que conduz ao layout proposto.*

## Identificação do espaço ocupado por tabela

|  |  |  |
| --- | --- | --- |
| Nome Tabela | Dimensão do Registo | Nº de Registos (inicial/final) |
| BuyingGroup | 95 | 2 |
| City | 109 | 3758 |
| UserData | 85 | 1 |
| Tables\_info | 118 | 96 |
| StockItem | 267 | 236 |
| State | 93 | 58 |
| Space\_used | 99 | 23 |
| Size | 41 | 44 |
| SalesTerritory | 43 | 10 |
| SaleHeader | 34 | 70510 |
| SaleDetails | 36 | 228276 |
| Promotion | 26 | 1 |
| PasswordResetToken | 54 | 1 |
| Package | 27 | 6 |
| Location | 29 | 402 |
| Errors | 181 | 6 |
| Employee | 65 | 19 |
| CustomerCategory | 47 | 7 |
| Customer | 85 | 402 |
| Country | 41 | 2 |
| Continent | 41 | 2 |
| Constraints\_info | 171 | 39 |
| Color | 37 | 9 |
| UserTable |  |  |
| PasswordResetToken |  |  |

## Especificação dos FIlegroups

|  |  |  |
| --- | --- | --- |
| Nome Filegroup | Tabelas associadas | Parâmetros |
| Primary | SaleHeader, SaleDetails, StockItem, Promotion, Employee, PasswordResetToken, UserData, Customer, Errors, SpaceUsed, TablesInfo, ConstraintsInfo | Size: 20 MB  Max size: 50 MB  File growth: 5 MB |
| WWI\_DATA | State, Country, Continent, SaleTerritory, City, Location, BuyingGroup, CustomerCategory, Color, Package, Size | Size: 4 MB  Max size: 8 MB  File growth: 2 MB |

## Schemas

|  |  |
| --- | --- |
| Nome | Descrição |
| SalesMgt | Informação relevante às vendas |
| readData | Dados que são apenas lidos e não escritos |
| customer | Informação relevante aos clientes |
| stock | Informação relevante aos produtos |
| auth | Informação relevante sobre a base de dados em si / informação sensível |

# Verificação da migração de dados

## Consultas sobre a base de dados original

## Consultas sobre a nova base de dados

# Programação

## Views

|  |  |
| --- | --- |
| Nome | Descrição |
| LastTablesData | Retorna a *view* das últimas modificações feitas às tabelas |
| LastConstraintsData | Retorna a *view* das últimas modificações feitas às *table constraints* |
| SalesData | Retorna a *view* com o valor calculado para cada venda |

## Functions

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | Atributos | Requisito | Descrição |
| dbo.fn\_countProductsInSale | @SaleID INT |  | Permite obter o número de produtos dentro de uma venda |
| dbo.fnGetSaleTotalPrice | @SaleID INT | R12# | Permite obter o valor monetário total para uma venda |
| Dbo.fn\_checkChiller | @SaleID INT,  @Chiller bit |  | Permite verificar se uma dada venda não possui items que são guardados em refrigador e em temperatura do ambiente ao mesmo tempo |
| Dbo.fn\_checkTime | @SaleID INT, |  | Permite verificar se um item foi enviado ao consumidor dentro do tempo previsto pelo item |

## Stored procedures

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | Atributos | Requisito | Descrição |
| dbo.sp\_removeStockItem | @saleID int  @itemID int  @removeSale nvarchar(50) | R11# | Permite remover um produto de uma venda |
| dbo.sp\_update\_product\_qty | @SalesDetailID int  @Qty int | R10# | Permite alterar a quantidade de um dado produto em uma venda |
| dbo.sp\_new\_SaleDetail | @SalesHeaderID int  @Product int  @Qty int  @CustomerID int | R09# | Permite adicionar um produto a uma sale |
| Dbo.sp\_newSaleHeader | @SalesPersonID int  @CityID int  @InvoiceDateKey date  @DeliveryDateKey date  @Profit int | R08# | Permite criar uma nova sale |
| dbo.sp\_addPromotion | @ItemID int  @Discount decimal(18,3)  @Start date  @End date | R07# | Permite adicionar uma promoção a um produto |
| dbo.sp\_removePromotion | @ItemID int | R07# | Permite remover uma promoção a um produto |
| dbo.sp\_createUser | @CustomerID int  @hash binary(64)  @mail nvarchar(255) | R04# | Permite criar um novo utilizador |
| dbo.sp\_editUser | @CustomerID int  @hash binary(64)  @mail nvarchar(255) | #R05 | Permite editar um utilizador |
| Dbo.sp\_removeUser | @UserID int | #RO5 | Permite remover um utilizador |
| Dbo.sp\_generate\_token | @UserID int | #R02 | Permite gerar um token para alterar a password |
| Dbo.sp\_password\_check | @mail nvarchar(255)  @hash varchar(30) |  | Checks if a certain password matches a user’s password |

## Triggers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome | Tipo | Tabela | Requisito | Descrição |
| Dbo.trg\_ChillerStockSale\_Insert | INSTEAD OF INSERT | Dbo.SaleDetails | R14# | Verifica se dentro de uma sale, todos os produtos são guardados no refrigerador ou nenhum é. |
| Dbo.trg\_LeadTimeDays\_Insert | INSTEAD OF INSERT | Dbo.SaleHeader | R13# | Verifica se o produto chegou ao consumidor dentro do tempo alotado |
| Dbo.trg\_ChillerStockSale\_Update | INSTEAD OF UPDATE | Dbo.SaleDetails | R14# | Verifica se dentro de uma sale, todos os produtos são guardados no refrigerador ou nenhum é. |
| Dbo.trg\_LeadTimeDays\_Update | INSTEAD OF UPDATE | Dbo.SaleHeader | R13# | Verifica se o produto chegou ao consumidor dentro do tempo alotado |

# Catálogo/Metadados

## Geradores

|  |  |  |
| --- | --- | --- |
| Nome | Atributos | Descrição |
| Dbo.sp\_generator | @table VARCHAR(32) @action char | Implementa o procedimento para inserir registos numa tabela |
|  |  |  |

## Monitorização

|  |  |  |
| --- | --- | --- |
| Nome | Atributos | Descrição |
| table\_update |  | Extraí o último update da data sobre as modificações do schema e compara-os à base de dados atual. Se houver modificações, gera entradas à tabela *TablesInfo* e *ConstraintsInfo* para manter um *log* de histórias de modificações |
| space\_used\_data |  | Gera entradas à tabela *SpaceUsed* para manter história de cada tabela na base de dados, os números de registos e uma estimativa do espaço ocupado |

# Índices

* 1. Views

|  |  |
| --- | --- |
| Nome | Descrição |
| citySales | Pesquisa o número de vendas por cidade |
| growthRate | Pesquisa a taxa de crescimento das vendas por ano |
| itemColor | Pesquisa os produtos por cor |

* 1. Índices

|  |  |  |
| --- | --- | --- |
| Designação | Tabela | Justificação/Consultas |
| City\_index | City(Name) | Torna a pesquisa de cidades mais rápida para ser usada em pesquisas de vendas por cidade |
| InvoiceDateKey\_index | SaleHeader(InvoiceDateKey) | Torna a pesquisa das vendas por ano mais rápida, que pode ajudar com comparações anuais |
| stockItem\_index | SaleDetails(StockItemID) | Torna as pesquisa dos produtos mais rápida que pode ajudar com o *performance* da bd em geral |

* 1. Otimização e Execução de Consultas

1. Backup e Recuperação
2. Segurança e Controlo de Acessos
   1. Níveis de acesso à informação
   2. Encriptação
3. Controlo de Concorrência
4. MongoDB

A fim de efetivamente recuperar os dados da versão *MongoDB* da base de dados *WWIGlobal,* foi decidido criar duas coleções distintas, *Sales* e *StockItems*. Para importar dados na nova base de dados *NoSQL,* uma série de consultas foram criadas e exportadas como ficheiros *JSON*. Existiam vários métodos para inserir dados para dentro da base de dados, mas escolhemos em utilizar a ferramenta do *MongoDB*, “*mongoimport”.* Estas exigia a especificação da *connection string*, do nome da nova coleção, a base de dados, o caminho para o ficheiro e a opção *“—jsonarray”* para os dados serem inseridos de foram eficiente.

# Descrição da Demonstração

## Script de demonstração

Sequência de execução de código (scripts, consultas, chamadas a SPs,…) que permita verificar o correto funcionamento da base de dados face aos requisitos.

exec sp\_createUser 1, N'teste123', 'ullasa@ukr.net'

exec sp\_generate\_token 1

exec sp\_editUser 1, 1, N'teste123', 'ullasa@ukr.net'

exec sp\_removeUser 1

exec space\_used\_data

exec table\_update

select dbo.fn\_countProductsInSale(4)

exec sp\_removeStockItem 4, 102, ‘no’

exec sp\_removeStockItem 1, 182, ‘yes’

select dbo.fnGetSaleTotalPrice(4)

exec sp\_update\_product\_qty 1, 2

exec sp\_new\_SaleDetail 1,2,3,4,15.00

exec sp\_addPromotion 1,15.00,'2022-11-21','2022-12-24'

exec sp\_removePromotion 1

exec sp\_createUser 1,2,'mail'

exec sp\_generate\_token 1

exec sp\_editUser 2,4,5,'novo'

exec sp\_removeUser 1

exec sp\_removeStockItem -1, 1, 'yes'

select \* from Errors

update SaleHeader set

DeliveryDateKey = '2013-02-25'

where SaleHeaderID = 5

exec sp\_new\_SaleDetail 1,72,1,2,15.00

# Conclusões

Concluímos que o esquema de dados antigo possuía bastantes erros de normalização e não se encontrava em estado para utilização empresarial. O processo de tratamento dos dados e criação da nova base de dados foi bastante trabalhosa o que reflete no mau estado da base dados no estado inicial.

A base de dados resultante admite-se bastante completa de funcionalidades para o cliente final. Ele poderá fazer diferentes operações na base de dados sem que o mesmo seja impedido por erros ou inconsistências no sistema.

Porém a base de dados ainda apresenta problemas de eficiência que possivelmente serão abordados nas entregas futuras do trabalho.