

**Relatório do Projeto de Data Warehouse**

**“Análise do serviço docente”**

**Realizado por:**

Catarina Rodrigues nº201501626;

Eduardo Palma nº201900054;

Nuno Melo nº201700465;

Ricardo Santos nº201700524;

**Docente:**Norberto Albino

**Unidade Curricular:**Data Warehouse

**Curso:** Licenciatura em Bioinformática, 2º ano

**Ano Letivo:** 2020/2021

**Instituição:** Escola Superior de Tecnologia do Barreiro

Índice

[1. Introdução 1](#_Toc59044326)

[1.1 Data Warehouse 1](#_Toc59044327)

[1.2 Modelo de Dados 1](#_Toc59044328)

[1.3 Modelo Dimensional 1](#_Toc59044329)

[1.4 Fases do Modelo Dimensional 2](#_Toc59044330)

[1.5 Matriz Bus 3](#_Toc59044331)

[1.6 Matriz Prioridades 3](#_Toc59044332)

[1.7 Processo de negócio 4](#_Toc59044333)

[1.8 Granularidade 4](#_Toc59044334)

[1.9 Descrição das dimensões 4](#_Toc59044335)

[1.10 Star Schema 6](#_Toc59044336)

[2. Bibliografia 10](#_Toc59044337)

Índice de Tabelas

[Tabela 1 - Matriz Bus 3](#_Toc59024367)

[Tabela 2 - Descrição das dimensões presentes no processo de negócio 5](#_Toc59024368)

Índice de Figuras

[Figura 1 - Matriz Prioridades 4](#_Toc61712423)

[Figura 2 - Star Schema 6](#_Toc61712424)

[Figura 3 - Tabela de dimensão Unidade Curricular 7](file:///C:\Users\nunop\Downloads\Relatorio_do_Projeto_de_Data_Warehouse_final%20(1).docx#_Toc61712425)

[Figura 4- Tabela de dimensão da Habilitação Académica 7](file:///C:\Users\nunop\Downloads\Relatorio_do_Projeto_de_Data_Warehouse_final%20(1).docx#_Toc61712426)

[Figura 5 - Tabela de Dimensão do Docente 8](file:///C:\Users\nunop\Downloads\Relatorio_do_Projeto_de_Data_Warehouse_final%20(1).docx#_Toc61712427)

[Figura 6 - Tabela de Dimensão do Regime Contratual 8](file:///C:\Users\nunop\Downloads\Relatorio_do_Projeto_de_Data_Warehouse_final%20(1).docx#_Toc61712428)

[Figura 7 - Tabela de Dimensão da Turma 8](file:///C:\Users\nunop\Downloads\Relatorio_do_Projeto_de_Data_Warehouse_final%20(1).docx#_Toc61712429)

[Figura 8- Tabela de Dimensão de Curso 9](file:///C:\Users\nunop\Downloads\Relatorio_do_Projeto_de_Data_Warehouse_final%20(1).docx#_Toc61712430)

[Figura 9 - Tabela de Dimensão da Data 9](file:///C:\Users\nunop\Downloads\Relatorio_do_Projeto_de_Data_Warehouse_final%20(1).docx#_Toc61712431)

[Figura 10 - Tabela de Factos da Análise da Docência 10](#_Toc61712432)

[Figura 11 – Esquema do Processo ETL 10](#_Toc61712433)

[Figura 12 - Tabela Auditoria com exemplos 12](file:///C:\Users\nunop\Downloads\Relatorio_do_Projeto_de_Data_Warehouse_final%20(1).docx#_Toc61712434)

[Figura 13 - Package Início 12](file:///C:\Users\nunop\Downloads\Relatorio_do_Projeto_de_Data_Warehouse_final%20(1).docx#_Toc61712435)

[Figura 14 - Variáveis criadas 13](#_Toc61712436)

[Figura 15 - Inserção Inicial da Auditoria 13](#_Toc61712437)

[Figura 16 - Configuração da Task “Inserção Inicial da Auditoria” 13](#_Toc61712438)

[Figura 17 – Ficheiro .txt com a inserção dos values na tabela Auditoria 14](file:///C:\Users\nunop\Downloads\Relatorio_do_Projeto_de_Data_Warehouse_final%20(1).docx#_Toc61712439)

[Figura 18 - Mapeamento dos parâmetros da task “Inserção Inicial da Auditoria” 14](#_Toc61712440)

[Figura 19 - Result Set da task “Inserção Inicial da Auditoria” 15](file:///C:\Users\nunop\Downloads\Relatorio_do_Projeto_de_Data_Warehouse_final%20(1).docx#_Toc61712441)

[Figura 20 - Progressão da task “Inserção Inicial da Auditoria” para a task “Execute Package DimDocente” 15](#_Toc61712442)

[Figura 21 - Configuração da task “DimDocente” 16](#_Toc61712443)

[Figura 22 - Relacionamento dos parâmetros da task “DimDocente” 16](file:///C:\Users\nunop\Downloads\Relatorio_do_Projeto_de_Data_Warehouse_final%20(1).docx#_Toc61712444)

[Figura 23 Variáveis da task "DimDocente" 17](file:///C:\Users\nunop\Downloads\Relatorio_do_Projeto_de_Data_Warehouse_final%20(1).docx#_Toc61712445)

[Figura 24 - Result Set da task "ContarRegistosIniciais" 17](file:///C:\Users\nunop\Downloads\Relatorio_do_Projeto_de_Data_Warehouse_final%20(1).docx#_Toc61712446)

[Figura 25 Configuração da task “Contar Registos Iniciais” 17](file:///C:\Users\nunop\Downloads\Relatorio_do_Projeto_de_Data_Warehouse_final%20(1).docx#_Toc61712447)

[Figura 26 - Progressão da task “Contar Registos Iniciais” para a task “Task Insert Auditoria Inicial” 18](#_Toc61712448)

[Figura 27 - Ficheiro .txt com o script para fazer a inserção das variáveis inicias na tabela auditoria 18](#_Toc61712449)

[Figura 28 - Configuração da task “Task Insert Auditoria Inicial” 18](file:///C:\Users\nunop\Downloads\Relatorio_do_Projeto_de_Data_Warehouse_final%20(1).docx#_Toc61712450)

[Figura 29 Configuração da task “Task Insert Auditoria Inicial” 18](file:///C:\Users\nunop\Downloads\Relatorio_do_Projeto_de_Data_Warehouse_final%20(1).docx#_Toc61712451)

[Figura 30 - Progressão da task “Task Insert Auditoria Inicial” para o data-flow "PDW\_DimDocente" 19](#_Toc61712452)

[Figura 31-Ole DB Source 19](#_Toc61712453)

[Figura 32-Colunas retiradas da tabela de funcionários 19](file:///C:\Users\nunop\Downloads\Relatorio_do_Projeto_de_Data_Warehouse_final%20(1).docx#_Toc61712454)

[Figura 33 Script utilizado para retirar os valores necessários da tabela Funcionários 20](file:///C:\Users\nunop\Downloads\Relatorio_do_Projeto_de_Data_Warehouse_final%20(1).docx#_Toc61712455)

[Figura 34-Slowly Changing Dimension 20](file:///C:\Users\nunop\Downloads\Relatorio_do_Projeto_de_Data_Warehouse_final%20(1).docx#_Toc61712456)

[Figura 35 Slowly Changing Dimension 20](file:///C:\Users\nunop\Downloads\Relatorio_do_Projeto_de_Data_Warehouse_final%20(1).docx#_Toc61712457)

[Figura 36- Contagem dos registos extraídos 20](file:///C:\Users\nunop\Downloads\Relatorio_do_Projeto_de_Data_Warehouse_final%20(1).docx#_Toc61712458)

[Figura 37- Configuração do tipo de Coluna para cada registo 22](#_Toc61712459)

[Figura 38- Configurações para as colunas tipo Historical 22](file:///C:\Users\nunop\Downloads\Relatorio_do_Projeto_de_Data_Warehouse_final%20(1).docx#_Toc61712460)

[Figura 39 Configurações para as colunas tipo Fixed e Changing 22](file:///C:\Users\nunop\Downloads\Relatorio_do_Projeto_de_Data_Warehouse_final%20(1).docx#_Toc61712461)

[Figura 40- Slowly Changing Dimension concluido com as contagens de tipo de coluna 23](#_Toc61712462)

[Figura 41- Contagem das alterações de tipo 23](file:///C:\Users\nunop\Downloads\Relatorio_do_Projeto_de_Data_Warehouse_final%20(1).docx#_Toc61712463)

[Figura 42--Contagem das alterações de tipo 1 24](file:///C:\Users\nunop\Downloads\Relatorio_do_Projeto_de_Data_Warehouse_final%20(1).docx#_Toc61712464)

[Figura 43- Configuração do OLE DB Destination 24](#_Toc61712465)

[Figura 44- Mapeamento das colunas de entrada com as da tabela de destino 25](#_Toc61712466)

[Figura 45- Conexão entre o data-flow e task "Contar Registos Finais" 25](#_Toc61712467)

[Figura 46- Configuração da task “Registos Finais” 26](file:///C:\Users\nunop\Downloads\Relatorio_do_Projeto_de_Data_Warehouse_final%20(1).docx#_Toc61712468)

[Figura 47- Result Set da task “Registos Finais” 26](file:///C:\Users\nunop\Downloads\Relatorio_do_Projeto_de_Data_Warehouse_final%20(1).docx#_Toc61712469)

[Figura 48- Ligação da task " Contar Registos Finais" à task "Updade Auditoria” 27](#_Toc61712470)

[Figura 49- Configuração da task "Update Auditoria" 27](#_Toc61712471)

[Figura 50- Script em formato .txt do update da tabela auditoria 28](file:///C:\Users\nunop\Downloads\Relatorio_do_Projeto_de_Data_Warehouse_final%20(1).docx#_Toc61712472)

[Figura 51- Variáveis que vão ser inseridas na tabela auditoria 28](file:///C:\Users\nunop\Downloads\Relatorio_do_Projeto_de_Data_Warehouse_final%20(1).docx#_Toc61712473)

# Introdução

## Data Warehouse

Data Warehouse é uma ferramenta de suporte ao processo analítico de uma organização. Esta ferramenta possibilita a análise de grandes volumes de dados, recolhidos dos sistemas transacionais OLTP (*Online Transaction Processing*).

## Modelo de Dados

Um modelo de dados é utilizado para organização dos dados de uma base de dados. Este modelo define um conjunto de conceitos para a representação de dados (por exemplo entidades, tabelas, atributos, etc.) e podem existir diferentes níveis de abstração de representação.

## Modelo Dimensional

No modelo dimensional podemos armazenar a informação com base na estrutura em estrela – o star-schema.

Esta estrutura tem como objetivo facilitar a obtenção das 3 caraterísticas fundamentais do DWH:

* Compreensão da informação pelo utilizador;
* Performance das consultas efetuadas;
* Resistência à mudança.

Esta estrutura possui ainda algumas características que a tornam mais adequada:

* Forma simples e simétrica – a organização por entidades de negócio facilita a compreensão;
* Melhoria da performance – a elevada desnormalização alivia a necessidade de uniões entre tabelas. A pré-agregação da informação é outro fator que contribui para a melhoria da performance;
* Capacidade de atualização – como as tabelas têm uma estrutura similar, é possível alterar o modelo (adicionar colunas ou tabelas) com uma quantidade de esforço reduzida.

## Fases do Modelo Dimensional

1. **Encontrar os processos de negócio** - é um conjunto de atividades ou tarefas estruturadas relacionadas que produzem um serviço ou produto específico para os seus clientes. Serve para identificar e categorizar os seus processos de negócio, utilizando a matriz BUS.
2. **Definir a granularidade da informação -** o grão é o menor nível da informação e é definido de acordo com as necessidades definidas no início do projeto. É determinado para cada Tabela de Factos, já que normalmente os Factos possuem informações e granularidades distintas. Deve utilizar-se o nível mais atómico possível, para maximizar as potencialidades do DWH.
3. **Definir as tabelas de dimensão** - tabelas que representam as entidades que participam no processo de negócio e que contêm os descritivos do negócio.
4. **Definir as tabelas de factos** – é oúltimo elemento a ser definido, contêm os atributos do processo de negócio que contribuam para a análise do que estamos a medir. Estabelecem a ligação com as tabelas de dimensão.

## Matriz Bus

A matriz Bus é a ferramenta utilizada para criar, documentar e comunicar a arquitetura do data warehouse. A matriz é uma tabela em que as linhas representam os processos diretamente associados aos fluxos de informação no centro de uma organização e as colunas representam as dimensões conforme.

Tabela - Matriz Bus

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Processos de negócio** | **Data** | **Entidade/Serviço** | **Docente** | **Funcionário** | **Aluno** | **Curso** | **Estatuto** | **Habilitação Académica** | **Regime Contratual** | **Unidade Curricular** | **Produto Transacionável** | **Turma** | **Total** |
| Análise do docente | x |  | x |  |  | x |  | x | x | x |  | x | **7** |
| Matrícula na instituição | x |  |  | x | x | x | x | x |  | x | x | x | **9** |
| Transação de produto/serviço | x | x |  | x | x |  | x |  |  |  | x |  | **6** |
| Gestão de infraestrutura | x | x | x |  |  |  |  |  |  |  | x |  | **4** |
| Aproveitamento académico | x |  | x |  | x | x | x |  |  | x |  | x | **7** |
| Promoção institucional | x | x | x |  | x | x |  |  |  |  |  |  | **5** |

## Matriz Prioridades

Permite definir a importância dos processos de negócio a serem implementados assim como a sua facilidade de implementação.



Figura - Matriz Prioridades

## Processo de negócio

Um processo de negócio é o conjunto de ações ou tarefas sequenciais e inter-relacionadas que transformam os inputs em outputs. O processo de negócio que nos foi atribuído foi a “Análise do serviço docente”, em que consiste analisar todas as tarefas desempenhadas pelos docentes, com base nos dados disponíveis.

## Granularidade

A granularidade definida para a nossa tabela de factos é elevada, dado que a informação utilizada tem um nível de detalhe baixo.

## Descrição das dimensões

Em seguida, vamos apresentar na tabela 2 um breve resumo de cada uma das dimensões utilizadas no processo de negócio “Análise do serviço docente”, no seu nível mais baixo de detalhe.

Tabela - Descrição das dimensões presentes no processo de negócio

|  |  |
| --- | --- |
|  | **Análise do serviço docente** |
| **Dimensão** | **Caracterização da dimensão** |
| Data | Permite registar a data das alterações temporais nos dados. |
| Docente | Dimensão que carateriza os docentes e regista a informação pessoal dos mesmos. |
| Habilitação Académica | Dimensão que regista as informações académicas para cada docente. |
| Regime Contratual | Dimensão que regista as horas estipuladas em contrato. |
| Unidade Curricular | Dimensão com as características das unidades curriculares. |
| Turma | Dimensão que regista o número de alunos. |
| Curso | Dimensão com as características dos cursos. |

Agora vamos explicar cada dimensão mais detalhadamente:

**Dimensão Data** - A dimensão Data foi criada de forma a conter todas as datas que aparecem nos dados dos sistemas operacionais. O carregamento da dimensão Data é efetuado apenas na primeira vez, após a criação da data warehouse ou quando esta tabela se encontra vazia.

**Dimensão Docente** – A dimensão Docente contém todas as informações pessoais relevantes dos docentes, que posteriormente irão ser importantes no nosso processo de negócio.

**Dimensão Habilitação Académica** – A dimensão Habilitação Académica foi criada para conter toda a informação relativa à formação académica do docente.

**Dimensão Regime Contratual** – A dimensão Regime Contratual contém os detalhes do contrato laboral de cada docente, as horas contratuais.

**Dimensão Unidade Curricular** – A dimensão Unidade Curricular contém todas as informações de cada uma das unidades curriculares, relacionadas com as unidades curriculares lecionadas pelos docentes.

**Dimensão Turma** – A dimensão Turma foi criada para armazenar toda a informação sobre cada turma e o total de alunos de cada uma delas, pertinente aos alunos de cada curso.

**Dimensão Curso** – A dimensão Curso contém todas as informações de cada curso.

## Star Schema

Um *Star Schema*é um tipo de modelo de dados de desenho de data Warehouse, e consiste em modelar os dados em tabelas dimensionais ligadas a uma tabela de factos. As **tabelas dimensionais**contêm as características de um evento. A **tabela de factos**armazena os factos ocorridos e as chaves para as características correspondentes, nas tabelas dimensionais.

Podemos observar na seguinte figura 2 como é geralmente representado este modelo.

Figura - Star Schema

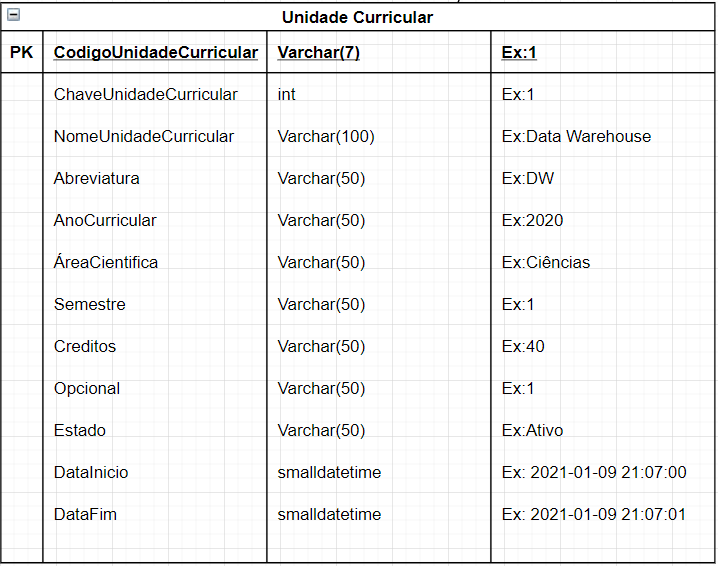
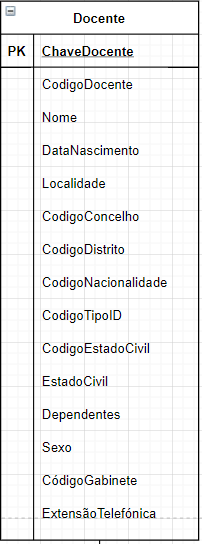


Figura - Tabela de dimensão Unidade Curricular



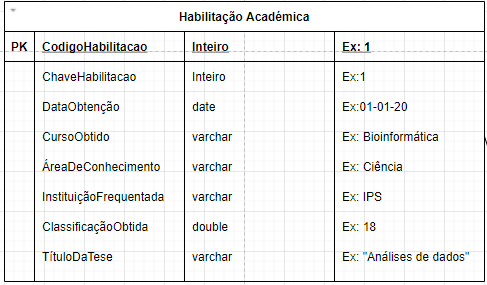


Figura - Tabela de dimensão da Habilitação Académica

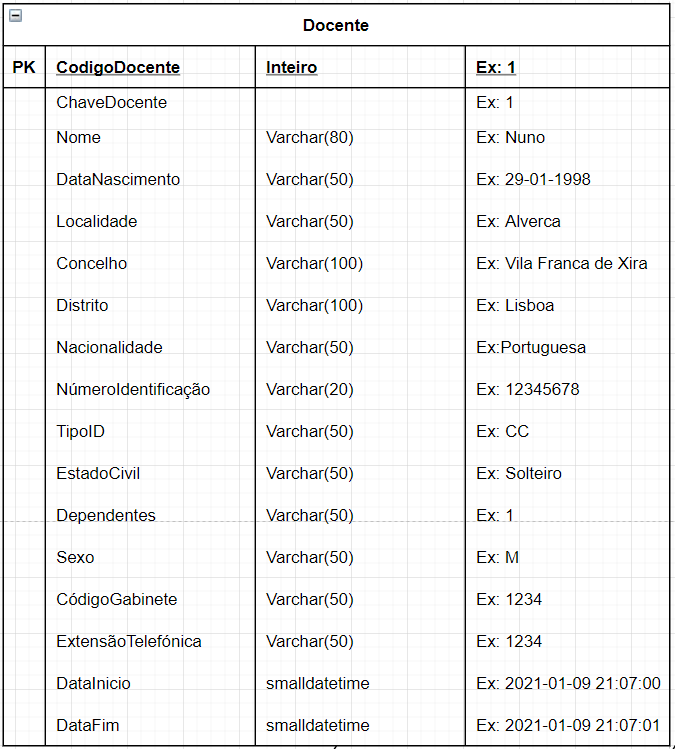


Figura - Tabela de Dimensão do Docente

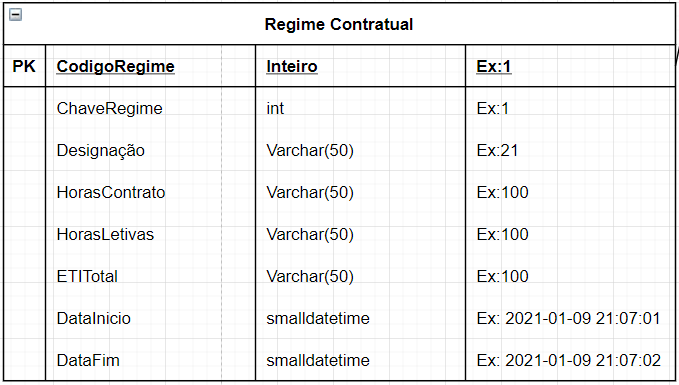


Figura - Tabela de Dimensão do Regime Contratual

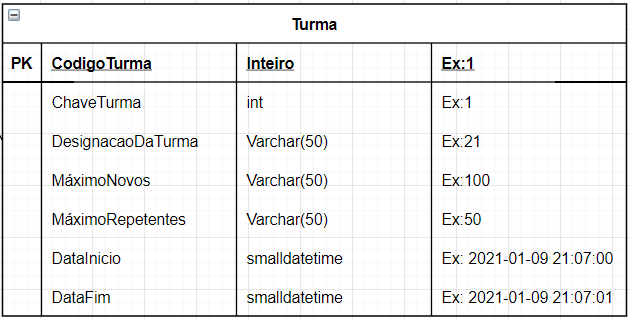


Figura - Tabela de Dimensão da Turma

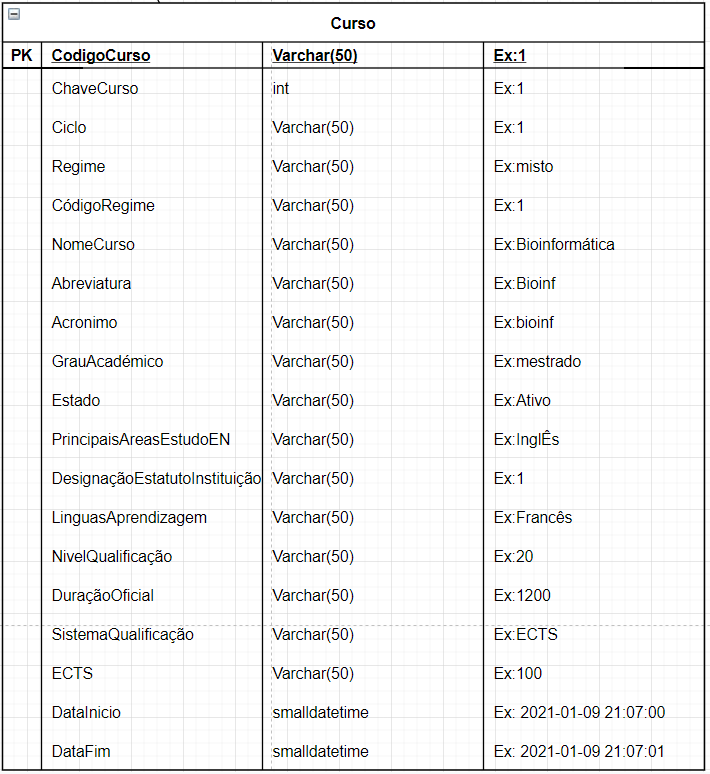


Figura - Tabela de Dimensão de Curso

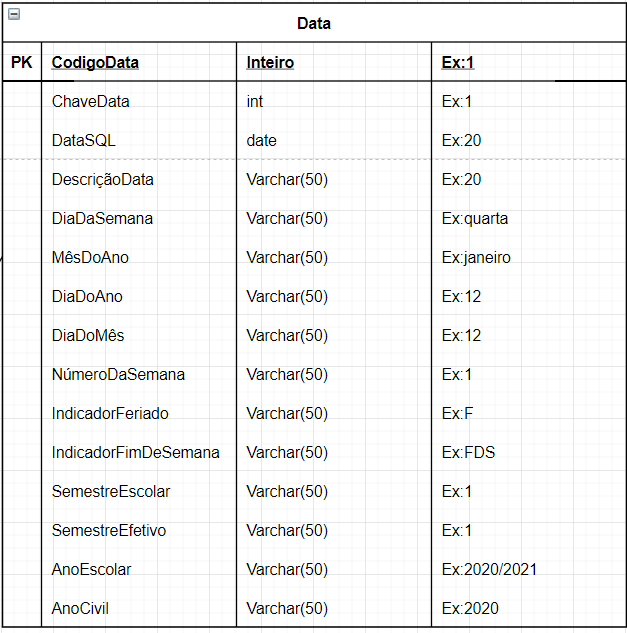


Figura - Tabela de Dimensão da Data

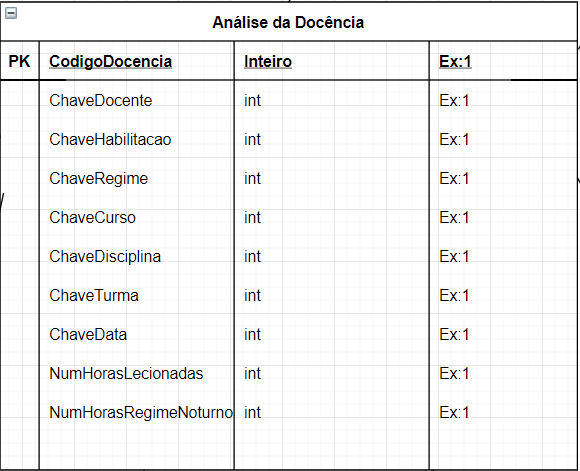


Figura - Tabela de Factos da Análise da Docência

# Auditoria para o Processo ETL

## O que é o Processo ETL?

É um processo onde são utilizadas ferramentas de software cuja função é a **extração** de dados de diversos sistemas**, transformação** desses dados conforme as regras dos negócios e por fim o **carregamento** dos dados para um sistema da organização, como mostra na figura 11.

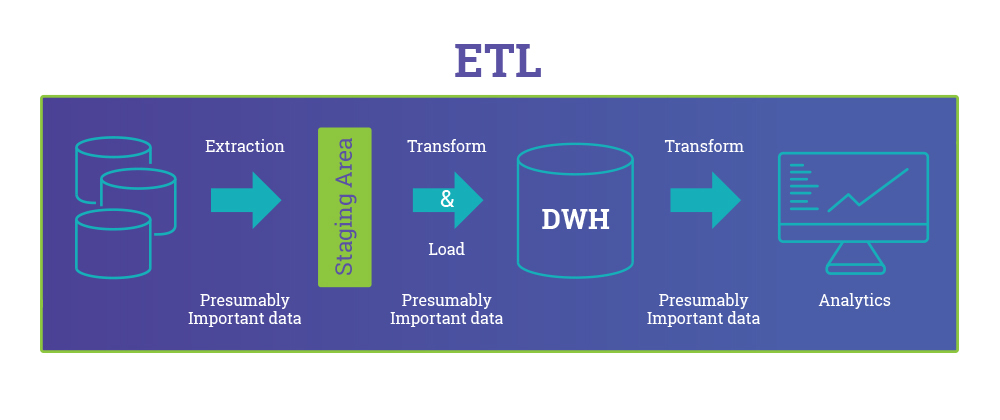


Figura – Esquema do Processo ETL

### Primeira Etapa – Extração

Nesta primeira fase procedemos á extração dos dados que pretendemos utilizar, extraídos de vários sistemas de origem com formatos ou organizações que podem ser diferentes ou então comuns (flat files).

A extração converte para um determinado formato para a entrada no processamento da transformação transferindo-os de seguida para o Data Warehouse.

### Segunda Etapa – Transformação

Nesta fase de transformação são aplicados um conjunto de regras ou funções aos dados anteriormente extraídos, onde pode existir muita ou pouca manipulação de dados. Esta manipulação é essencial pois irá melhorar a qualidade dos dados e consolidá-los.

Em alguns casos podem ser necessárias algumas transformações como por exemplo a junção de dados provenientes de várias fontes, a seleção de apenas algumas colunas e a tradução de valores codificados.

### Terceira Etapa – Carregamento

Esta última etapa consiste em estruturar e carregar os dados já alterados de acordo com os requisitos exigidos da empresa dentro da camada de apresentação seguindo o modelo *star schema*.

Alguns DW podem atualizar as informações existentes semanalmente, enquanto que outros podem adicionar dados novos de hora em hora. A latência e o alcance de reposição ou acréscimo constituem opções de projeto estratégicas que dependem do tempo disponível e das necessidades de negócio

## Tabela Auditoria

Na tabela auditoria é efetuado um rastreamento de qualquer operação que ocorre nos dados de cada uma das tabelas de dimensões, ou seja, sempre que um dado é inserido, bem ou mal, obter-se-á um registo nesta tabela.

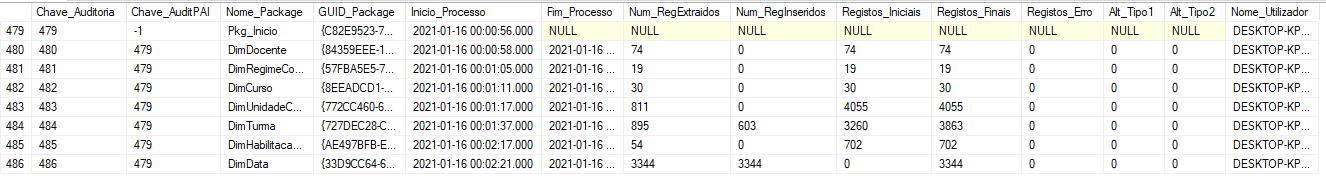


Figura - Tabela Auditoria com exemplos

# Explicação do Processo ETL

## *Package* Início

O *Package* Início, Figura 13, contém várias *tasks* (tarefas) que irão inicializar os *packages* que posteriormente irão adicionar os valores da DB(Base de Dados) SI\_ESTBarreiro nas nossas tabelas de dimensões e na tabela de auditoria na DB CatarinaNunoRicardoEduardo.

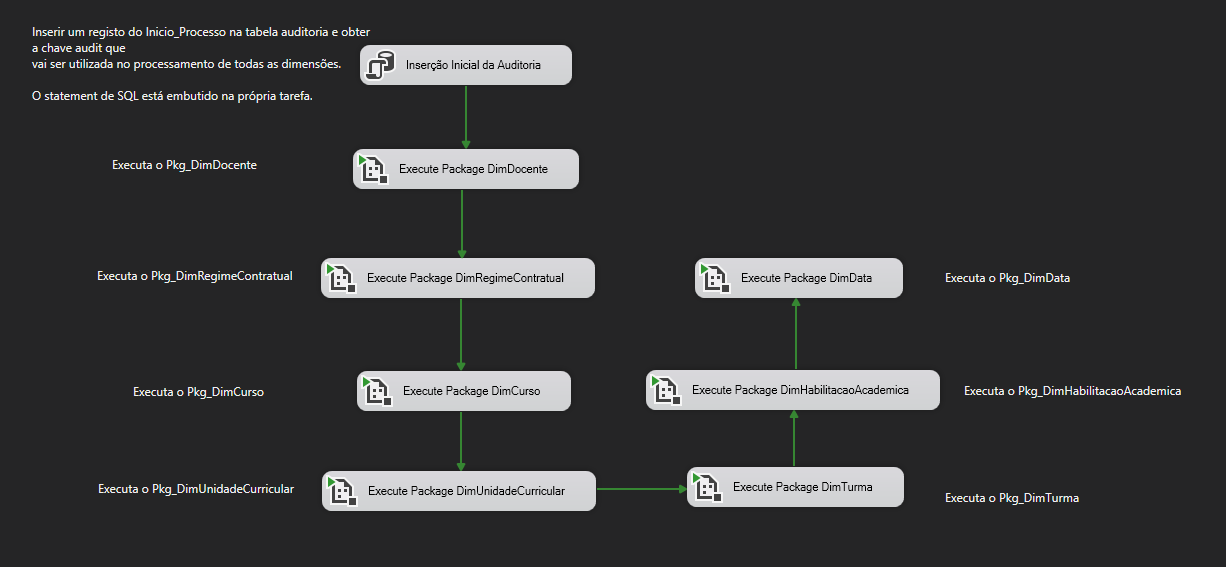


Figura - Package Início

Criou-se também várias variáveis que vão ser utilizadas ao longo do *package*, Figura 14, como por exemplo a Chave\_Auditoria que vai servir de indicador para cada auditoria feita e o GUID, que vai servir de indicador específico para cada *package* que vai se executado.

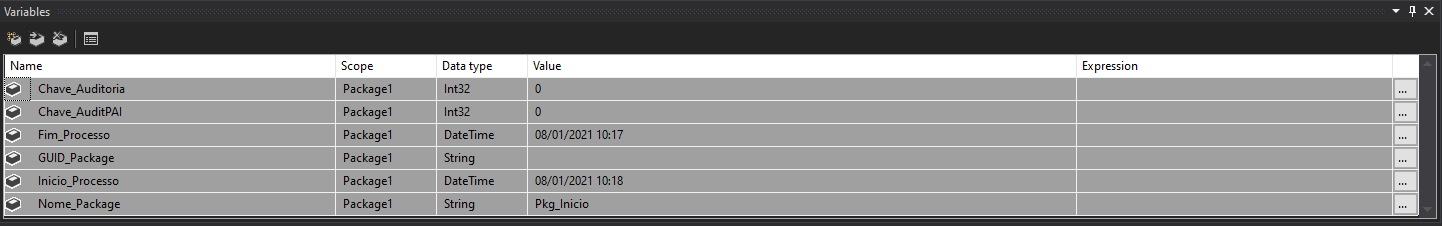


Figura - Variáveis criadas

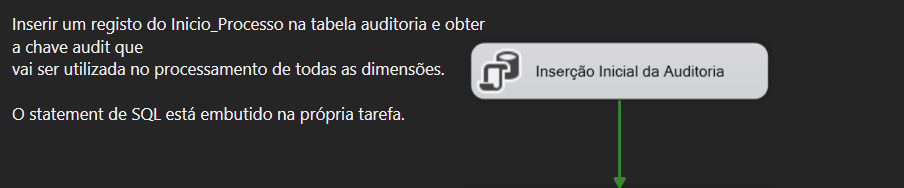


Figura - Inserção Inicial da Auditoria

Primeiro criou-se uma *task* do tipo *execute SQL* com o nome “Inserção Inicial da Auditoria” com a conexão do tipo ADO.Net e conectada á nossa DB que vai receber um ficheiro *.txt*.

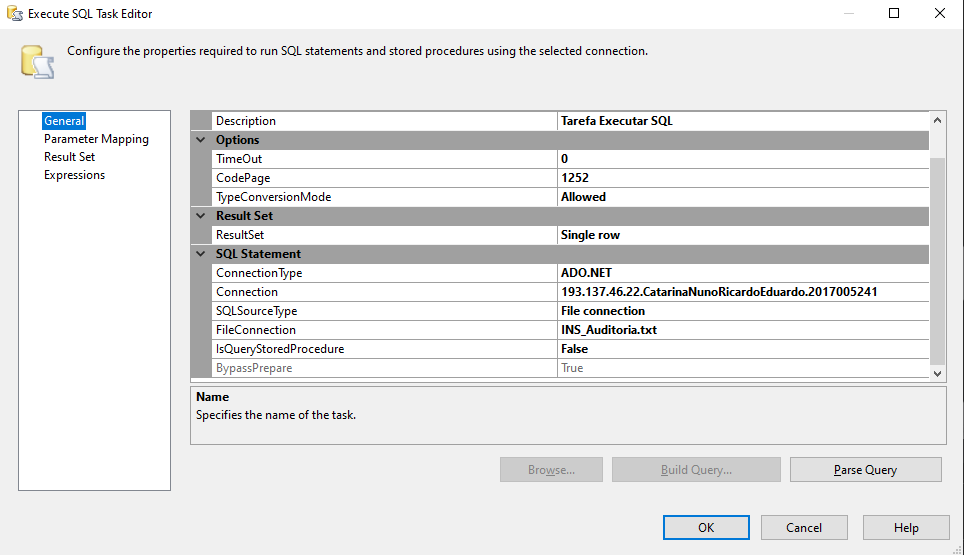


Figura - Configuração da Task “Inserção Inicial da Auditoria”

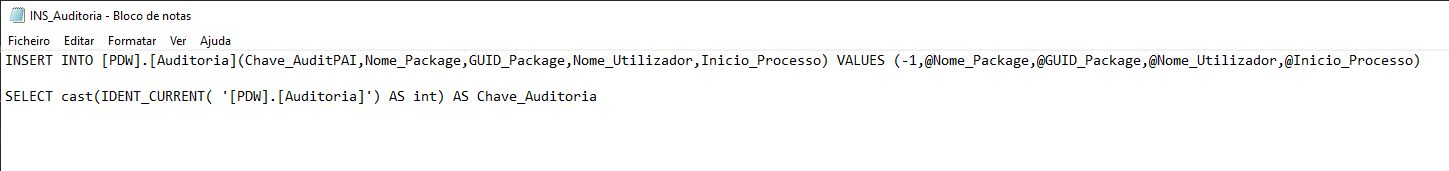
No *task* “Inserção Inicial da Auditoria” inseriu-se na tabela Auditoria os valores da Chave\_AuditPAI, Nome\_Package, GUID\_Package, Nome\_Utilizador e do Inicio\_Processo, que foram definidos nas variáveis, exceto a Chave\_AuditPAI que se forneceu um valor de -1, através do ficheiro *.txt*, como demonstra a figura 17.

Figura – Ficheiro .txt com a inserção dos values na tabela Auditoria

Depois procedeu-se ao mapeamento dos parâmetros onde se colocou os valores anteriormente referidos nas suas variáveis corretas como demonstra a figura 18.

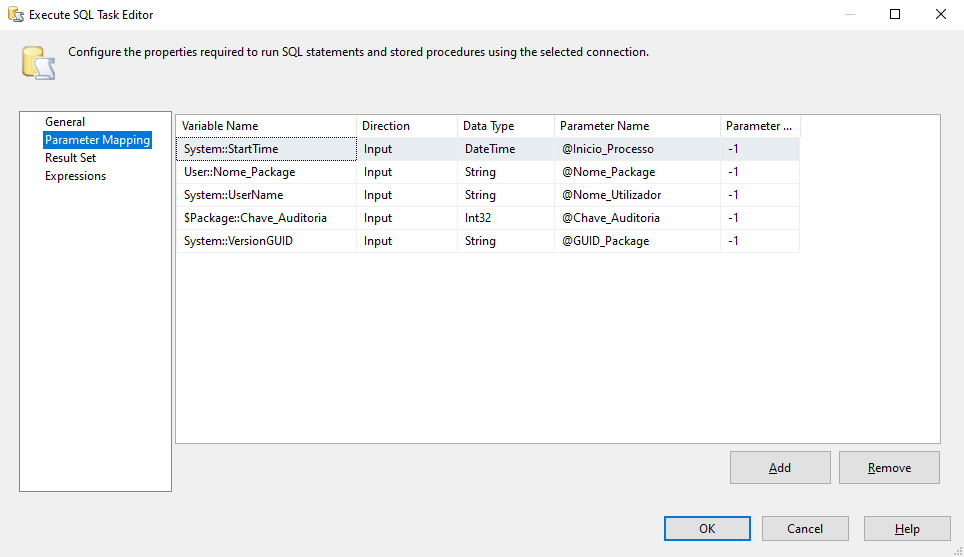


Figura - Mapeamento dos parâmetros da task “Inserção Inicial da Auditoria”

Quando a variável utiliza o prefixo *System* quer dizer que se usa dados do próprio sistema, ou seja, para a data de início do processo, *UserName* e *VersionGUID*, não foi necessário fornecer nenhum valor pois foram preenchidos automaticamente. Quando se usa o prefixo *User* quer dizer que lhe foi atribuído um valor pelo utilizador e quando é empregue o prefixo *$Package* utiliza-se uma variável que está sempre definido ao longo do *package*, não sendo necessária criá-la quando se cria uma *task.*

Ao criar-se esta *task* definiu-se também que esta terá um *result set*, como se pode verificar na figura 19, que será a Chave\_Auditoria, como mostra a figura 18, que será utilizada para que todas a *tasks* subsequentes dentro deste *package* terão como Chave\_AuditPai o valor desta.

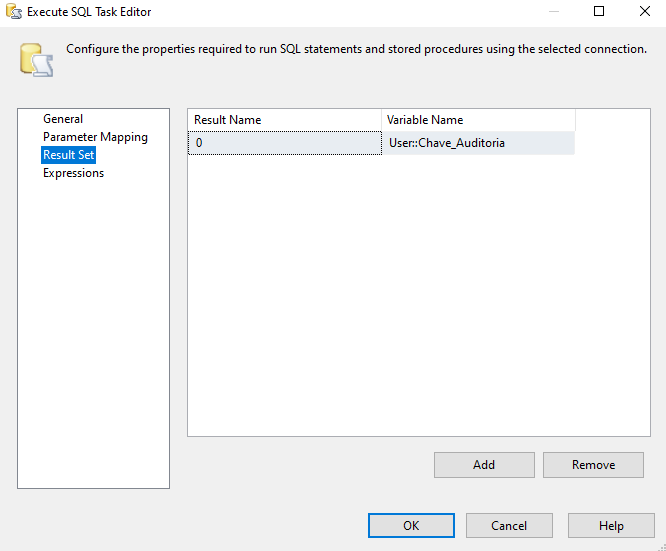


Figura - Result Set da task “Inserção Inicial da Auditoria”

## Tabelas de Dimensão

Após se dar início à auditoria pode-se prosseguir à inserção de dados. Como mostra a figura 20, logo após a *task* “Inserção Inicial da Auditoria” vai-se dar início à *task* “Execute Package DimDocente” que, como o nome indica, vai executar o *package* DimDocente.

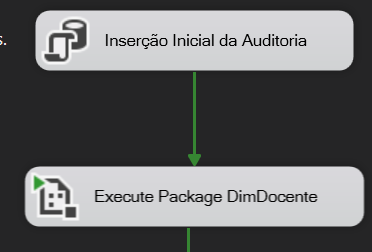


Figura - Progressão da task “Inserção Inicial da Auditoria” para a task “Execute Package DimDocente”

Quando se cria esta *task* do tipo execute *Package* vai-se referenciar o *package* escolhido, neste caso foi referenciado o DimDocente, como mostra a figura 21.

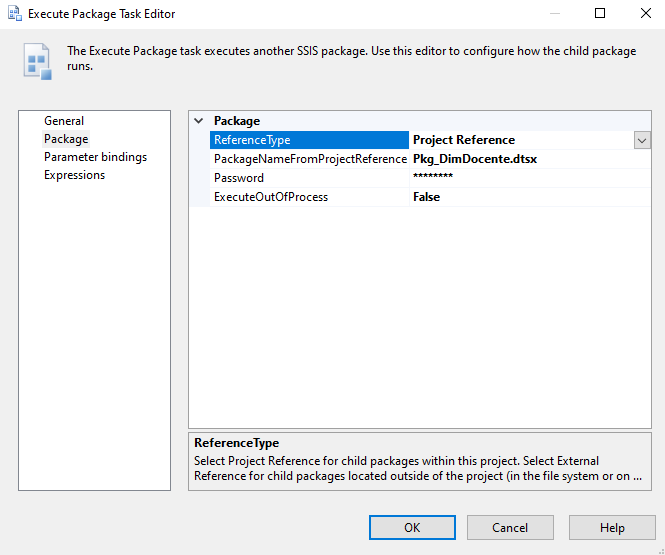


Figura - Configuração da task “DimDocente”

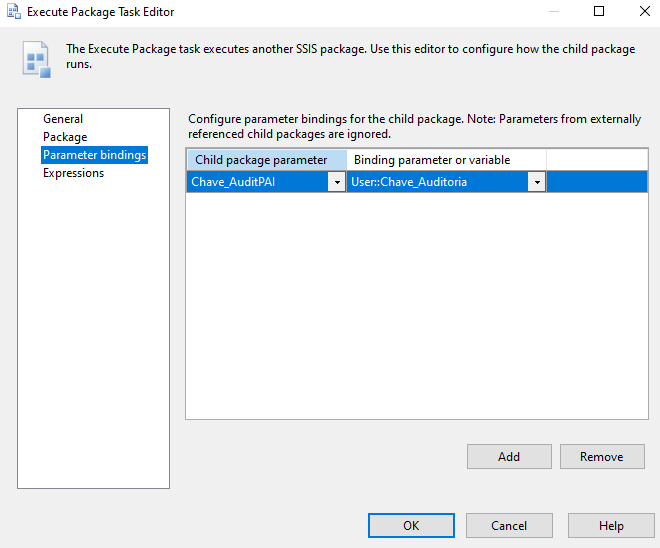
Depois procedeu-se ao relacionamento dos parâmetros onde definiu-se a *Child package* e o parâmetro de ligação que, como foi referido anteriormente vai fazer com que a Chave\_AuditPai tenha o valor da Chave\_Auditoria para que todas as *tasks* que vão ser executadas possam ser relacionadas à auditoria em específico, como mostra a figura 22.

Figura - Relacionamento dos parâmetros da task “DimDocente”

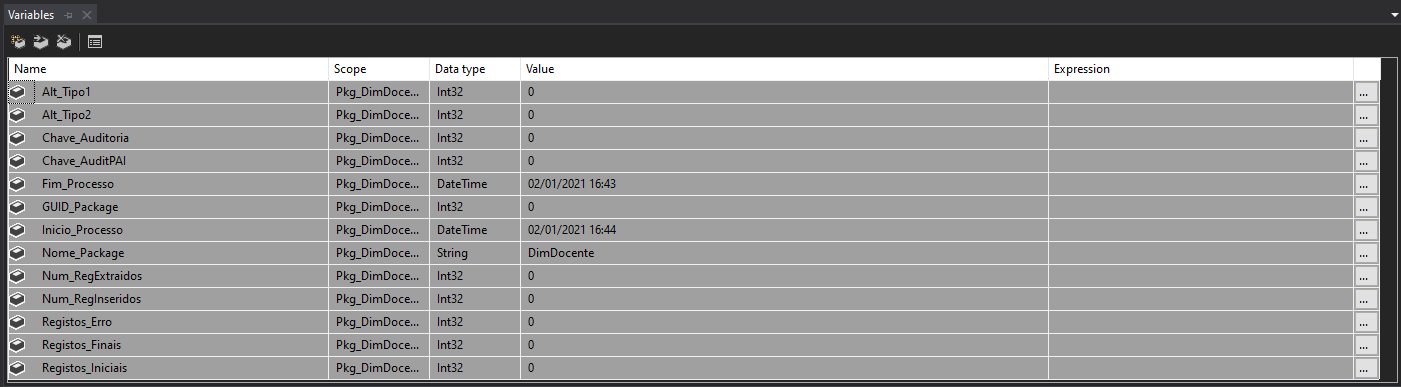
Após a *task* estar criada criou-se as variáveis que iram ser utilizadas, como demonstra a figura 23.

Figura Variáveis da task "DimDocente"

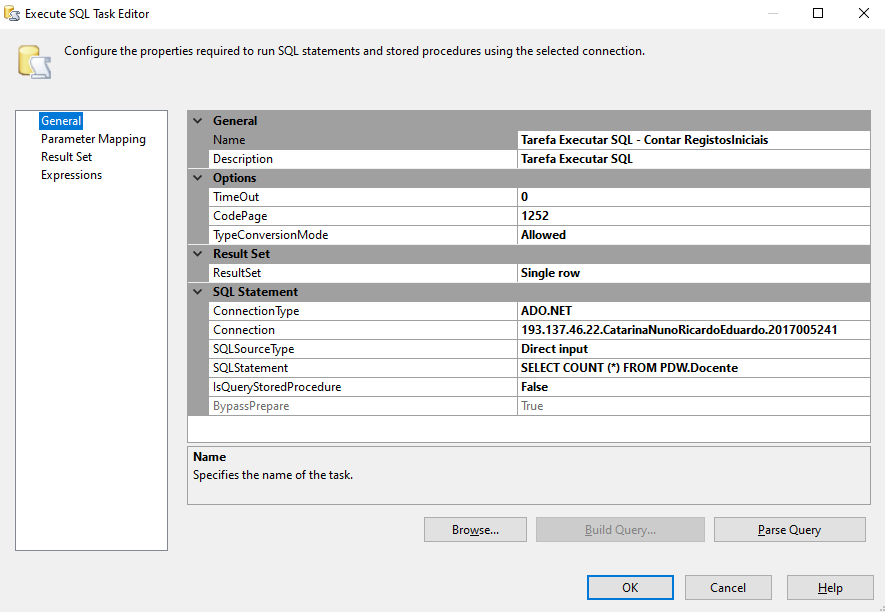
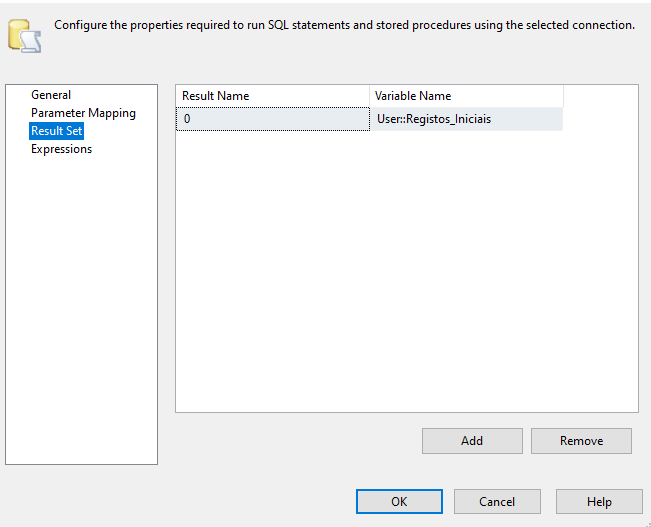
A primeira *task* que se criou foi do tipo *execute* SQL, com o nome “Contar Registos Iniciais” tipo de conexão ADO.Net e conectada à nossa DB CatarinaNunoRicardoEduardo, onde se vai efetuar uma contagem dos registos na tabela Docente para se saber quantos registos iniciais estão presentes, como mostra a figura 26, sendo este valor colocado numa variável num *result set*, referencia á figura 24.

Figura - Result Set da task "ContarRegistosIniciais"

Figura Configuração da task “Contar Registos Iniciais”

De seguida criou-se uma outra *task* do mesmo tipo e com a mesma conexão, figura 28, onde se vai, a partir de um ficheiro *.txt*, referencia à figura 26 e figura 27, vai introduzir os valores das variáveis presentes na figura 29 na tabela auditoria.

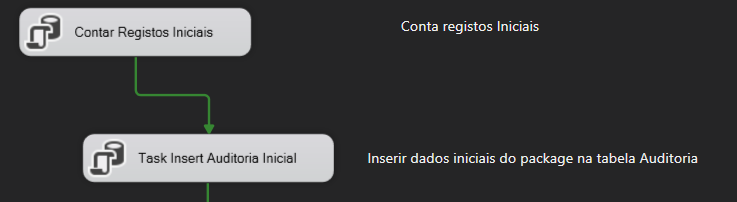


Figura - Progressão da task “Contar Registos Iniciais” para a task “Task Insert Auditoria Inicial”

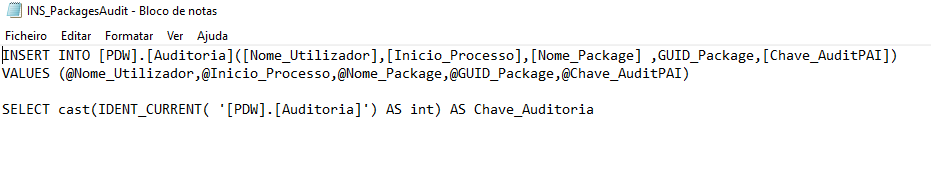


Figura - Ficheiro .txt com o script para fazer a inserção das variáveis inicias na tabela auditoria

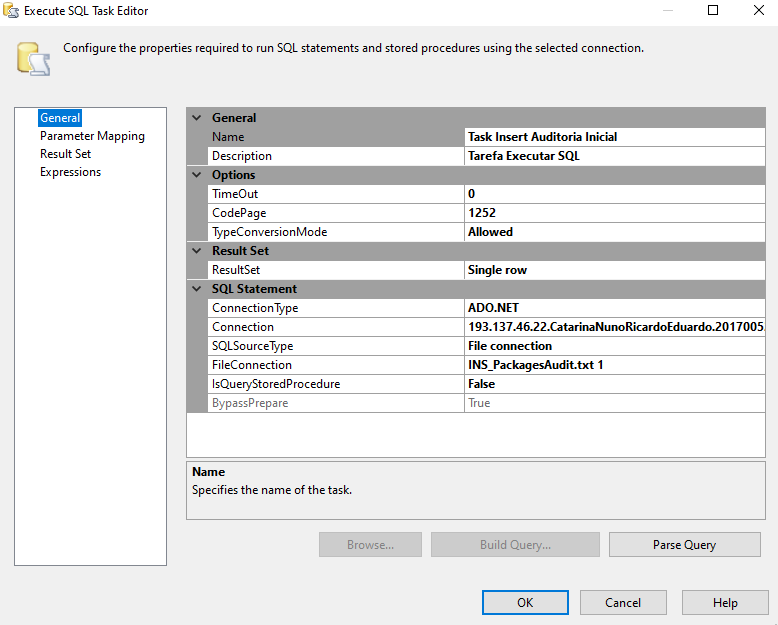
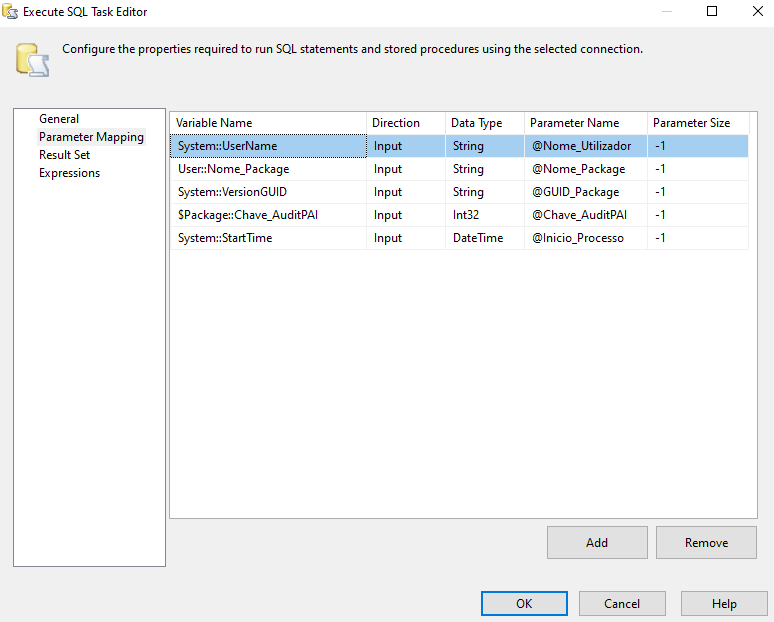


Figura - Configuração da task “Task Insert Auditoria Inicial”

Figura Configuração da task “Task Insert Auditoria Inicial”

De seguida criou-se um *data-flow como mostra a figura 30.*

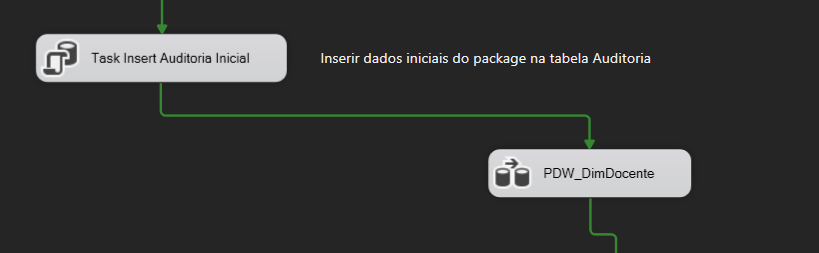


Figura - Progressão da task “Task Insert Auditoria Inicial” para o data-flow "PDW\_DimDocente"

Dentro do *data-flow* criou-se um OLE DB fonte, figura 31, que se conectou à DB SI\_ESTBarreiro onde se retirou os dados da tabela Funcionário, recorrendo a um script, figura 32 e figura 33.

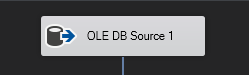


Figura -Ole DB Source

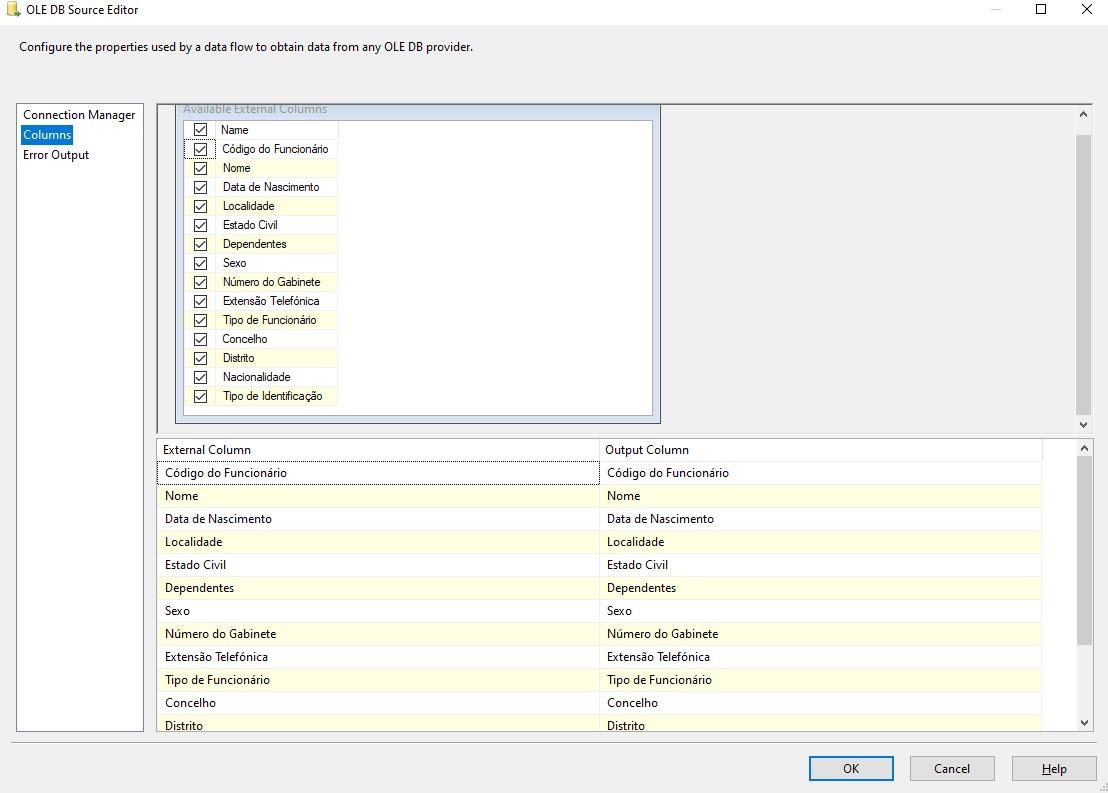


Figura -Colunas retiradas da tabela de funcionários

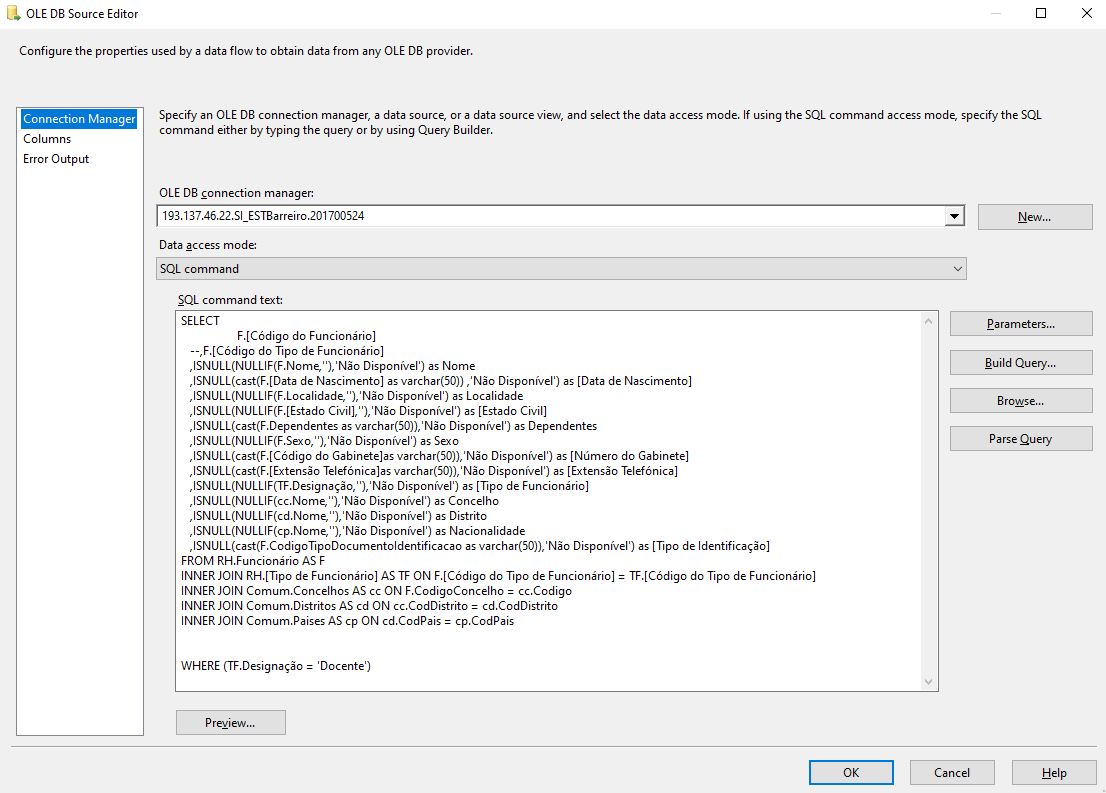


Figura Script utilizado para retirar os valores necessários da tabela Funcionários

Depois utilizou-se um *multicast* para fazer uma contagem dos valores extraídos que iram para um ficheiro à parte, figura 36, e também para aplicar uma *Slowly Changing Dimension (SCD)*, figura 34.

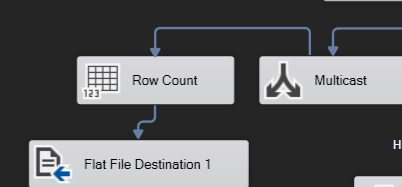


Figura -Slowly Changing Dimension

Figura Slowly Changing Dimension

Figura - Contagem dos registos extraídos

Ao inicializar-se o SCD foi necessário atribuir a que tabela esta iria colocar os nossos valores foi escolhida a tabela Docente do nosso DB pessoal. Para além disso foi também necessário escolher a *business key* a que foi atribuída á coluna Código de Funcionário, figura 35.

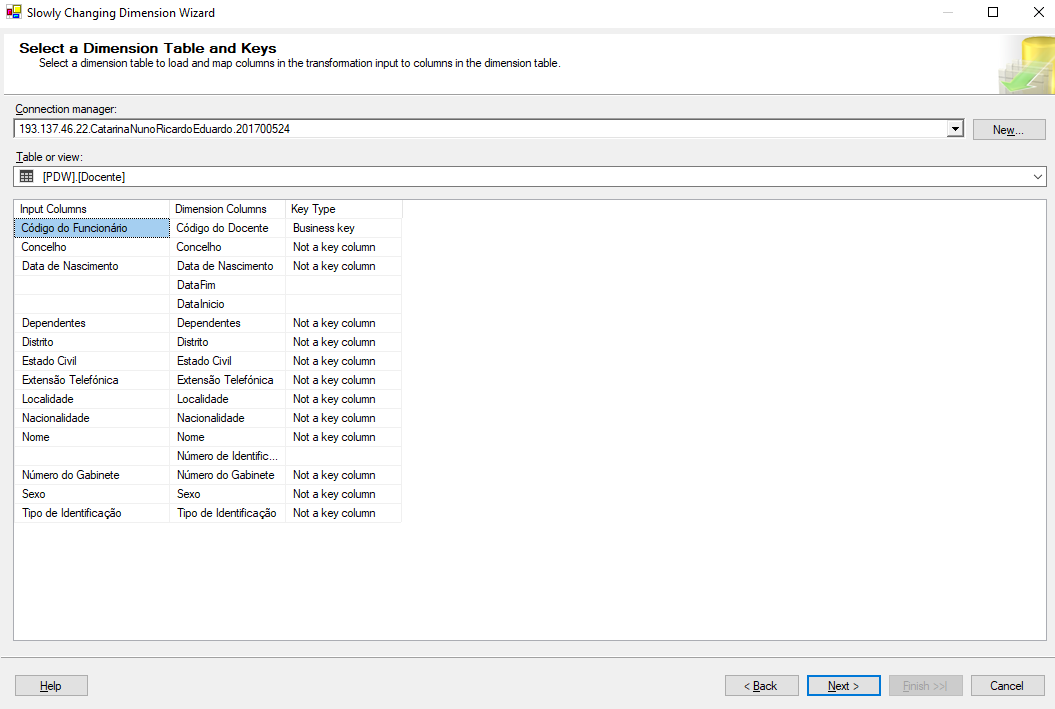


Figura 35- Atribuição da business key na Slowly Changing Dimension

Depois é necessário definir o tipo atributo de cada coluna, sendo o tipo *Fixed* valores que nunca mudam, *Changing* valores que devem de ser atualizados por cima dos valores antigos, descartando estes, e por fim *Historical* que são como os *changing* mas mantêm um registos dos valores antigos, sendo estes marcados como antigos, como demonstra a figura 37.

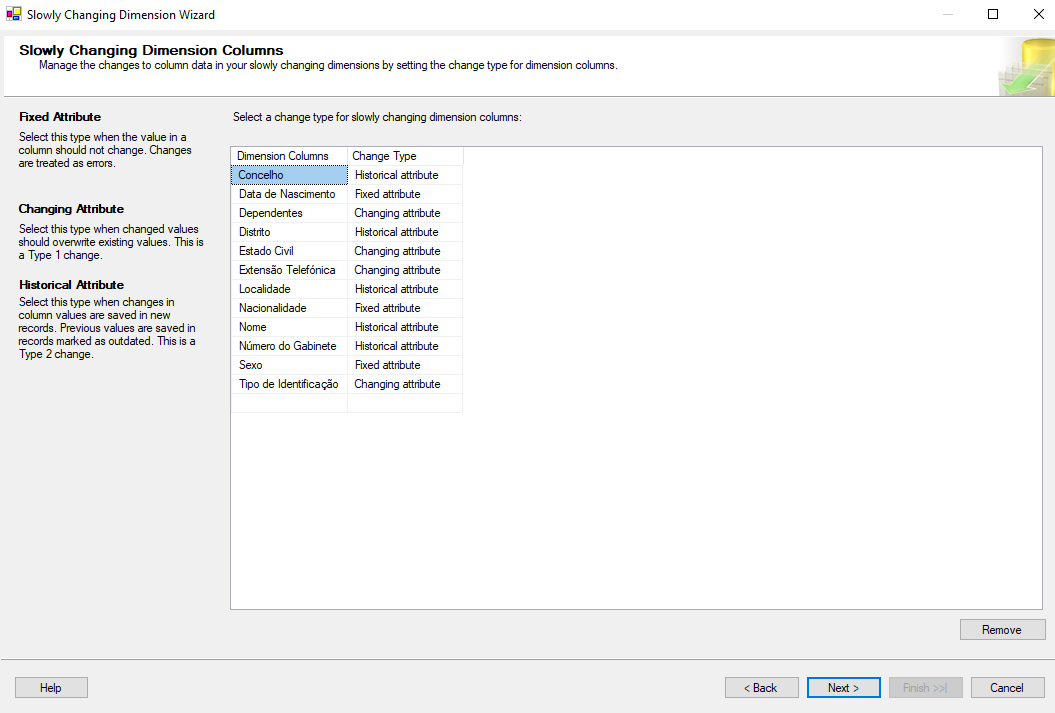


Figura - Configuração do tipo de Coluna para cada registo

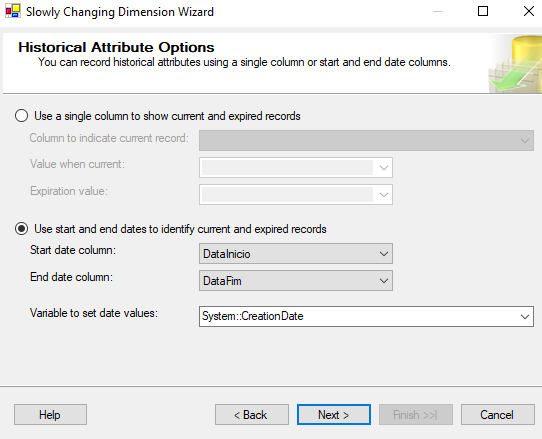
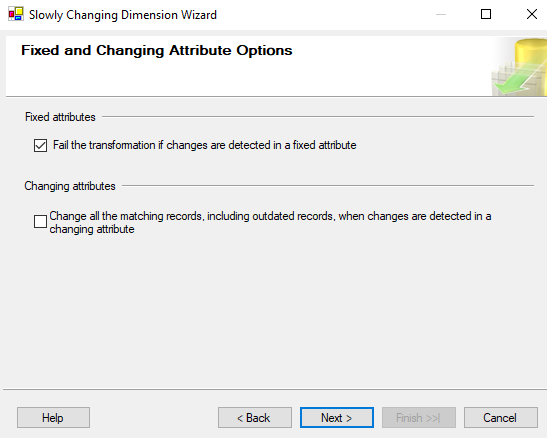
Após se ter feito a atribuição dos tipos de coluna, é necessário definir o que acontece caso se encontre alterações nas colunas para todos os tipos, como exemplifica as figuras 38 e 39.

Figura - Configurações para as colunas tipo Historical

Figura Configurações para as colunas tipo Fixed e Changing

Assim que se conclui o SCD, este devolve nos uma figura parecida á figura 39. As únicas alterações efetuadas foi a inclusão de dois *row-counts* para contar o número de alterações do tipo 1 (Changing) e do tipo dois (Historical), como evidencia as figuras 40, 41 e 42.

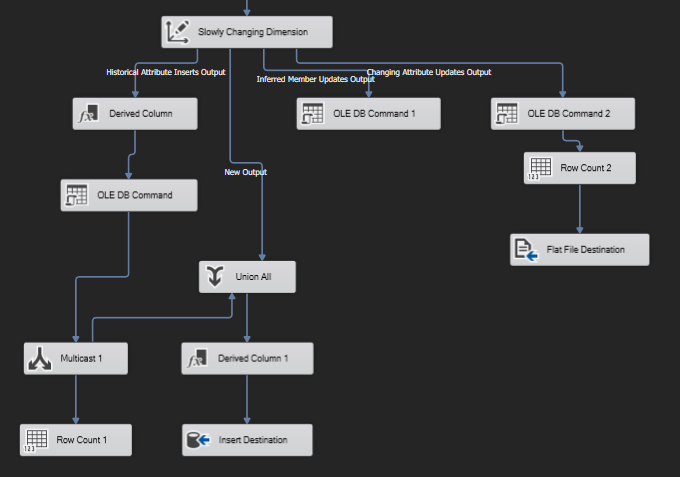


Figura - Slowly Changing Dimension concluido com as contagens de tipo de coluna

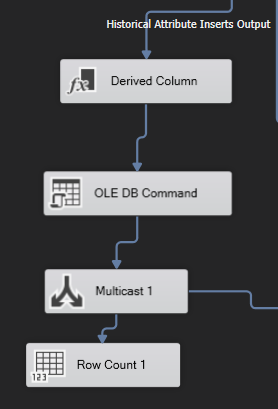


Figura - Contagem das alterações de tipo

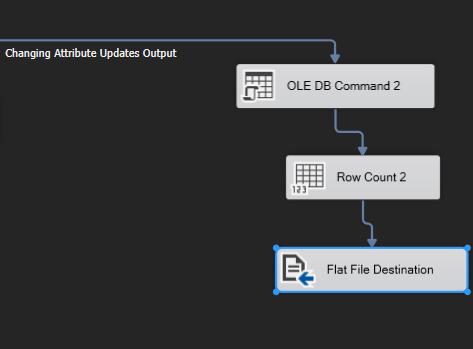


Figura --Contagem das alterações de tipo 1

Apesar do SCD fornecer um OLE DB *Destination* é preciso ainda conferir-lhe a DB e tabela corretas. Neste caso foi escolhida a tabela Docentes na nossa DB pessoal, como mostra a figura 43.

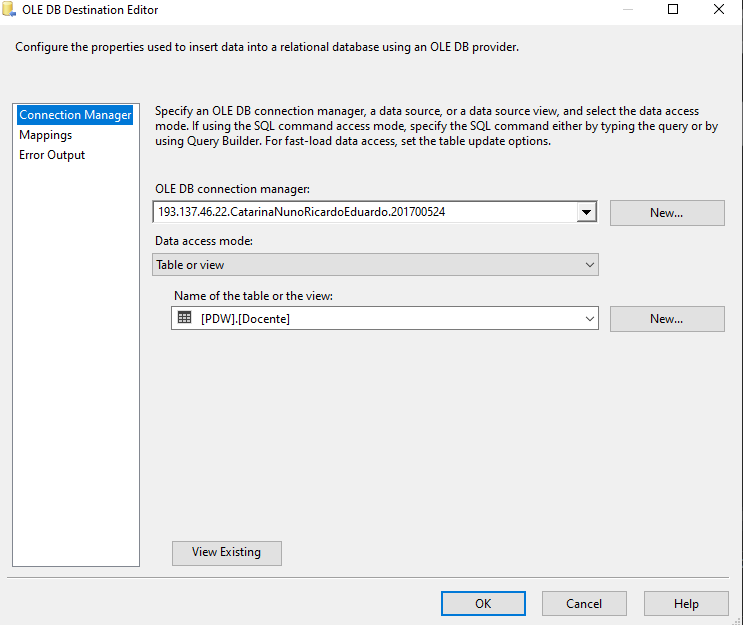


Figura - Configuração do OLE DB Destination

Depois realizou-se o mapeamento correto das colunas que o SCD fornece com as da tabela Docente, como exemplifica a figura 44.

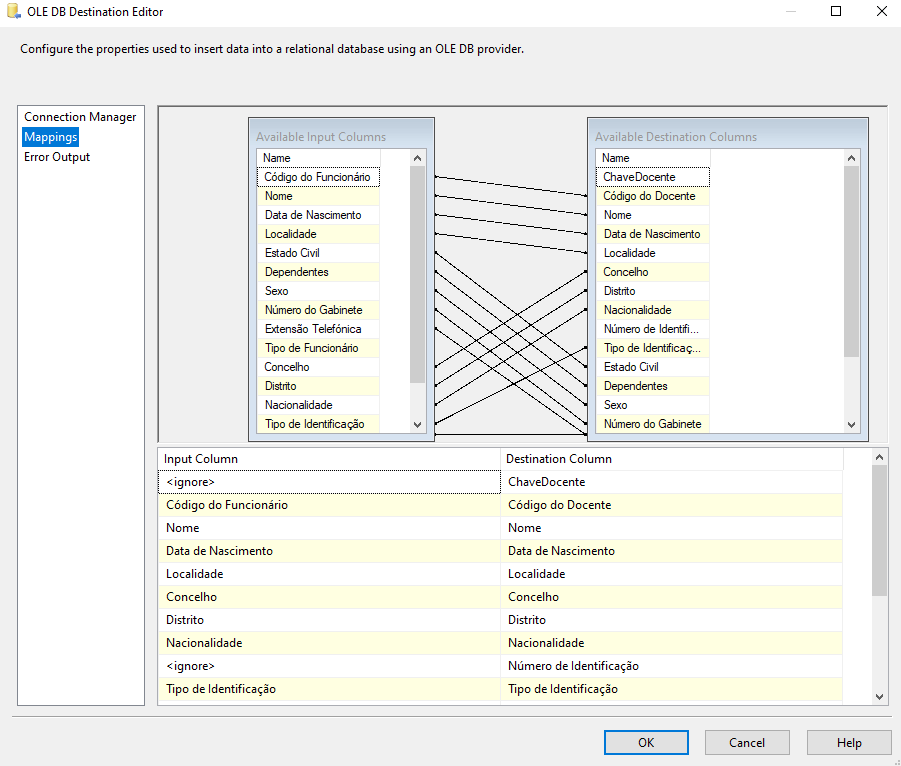


Figura - Mapeamento das colunas de entrada com as da tabela de destino

E com isto termina-se o *data-flow*. De seguida criou-se no *control-flow* uma nova task da mesma forma que a da contagem de registos iniciais, mas que vai contar os registos inseridos na tabela destino pelo *data-flow* e vai ter como *result set* os registos finais, figuras 45,46 e 47.

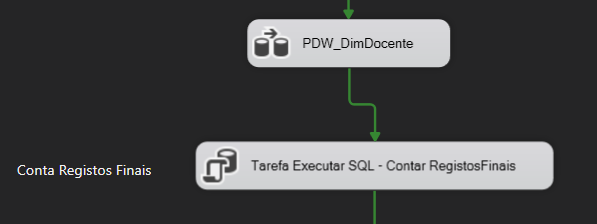


Figura - Conexão entre o data-flow e task "Contar Registos Finais"

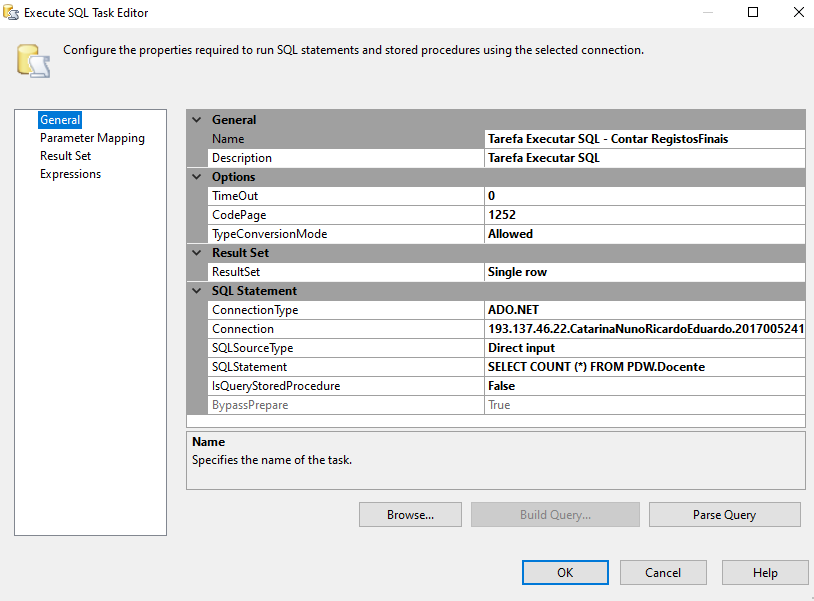


Figura - Configuração da task “Registos Finais”

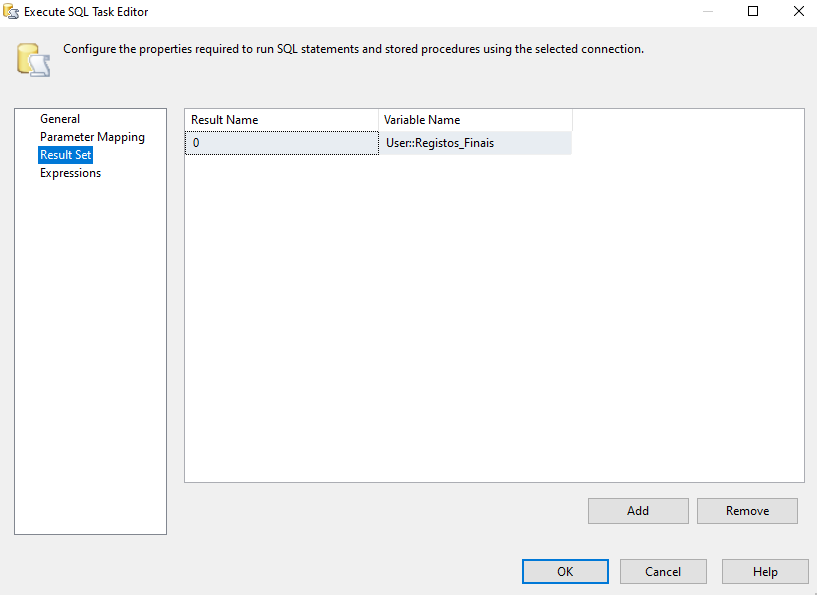


Figura - Result Set da task “Registos Finais”

Após a contagem estar realizada é necessário fazer um update á tabela auditoria, por isso cria-se uma nova task *execute* SQL para o fazer, figura 48.



Figura - Ligação da task " Contar Registos Finais" à task "Updade Auditoria”

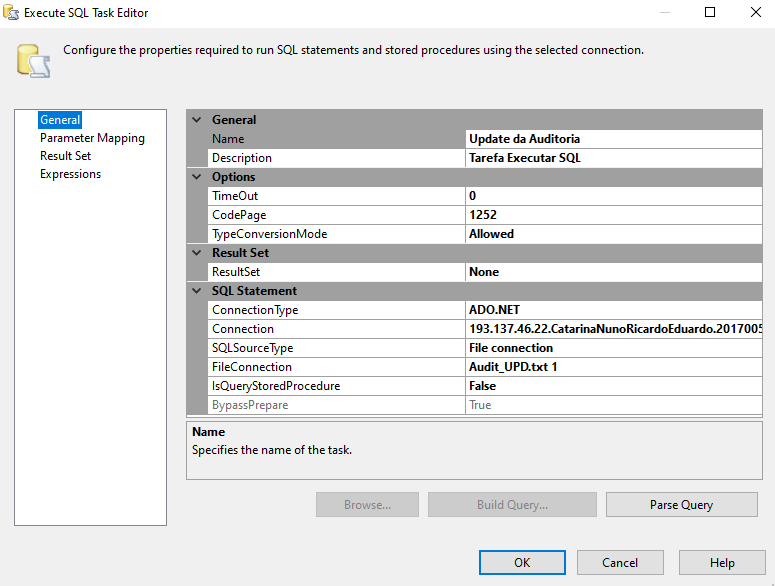
Esta task vai estar conectada á DB CatarinaNunoRicardoEduardo, figura 49, e vai correr um script de um ficheiro .txt, figura 50, que vai fazer a inserção dos valores das variáveis, figura 51.

Figura - Configuração da task "Update Auditoria"

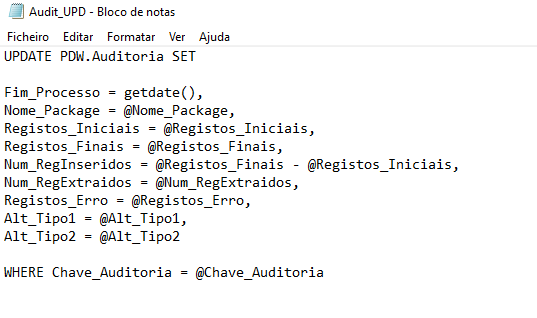


Figura - Script em formato .txt do update da tabela auditoria

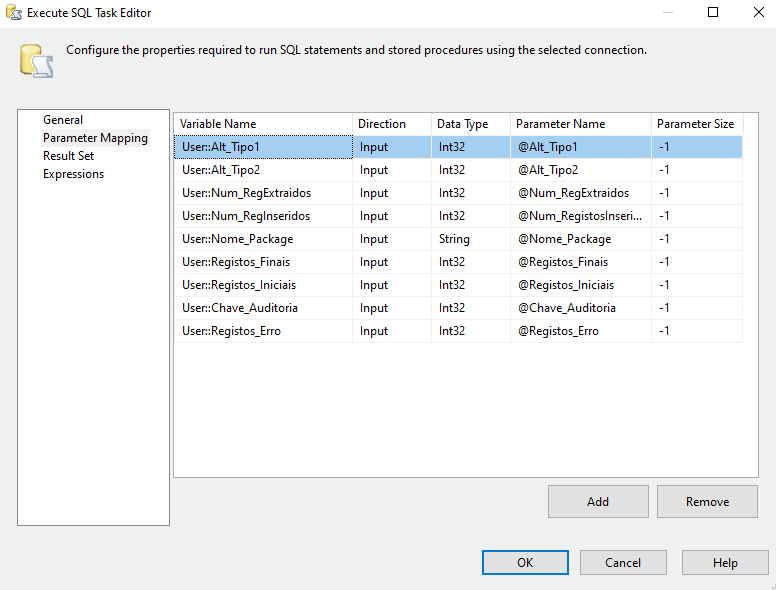


Figura - Variáveis que vão ser inseridas na tabela auditoria

Assim concluímos a *task* DimDocente. Todas as outras *tasks* foram criadas de igual modo apenas mudando os valores e as tabelas.

## Tabela de Factos

A tabela *StagingTable\_AnáliseDaDocencia* tem como objetivo carregar a tabela de factos através de *lookups* que verificam a conexão entre o nível operacional e a *star schema*.

Carregamos então a *staging table* com as chaves do nível operacional de modo a referenciá-las para subirmos e associarmos ao nível das *surrogate keys.*

Começámos por fazer um SQLStmt para mais tarde fazer as *views*, figura 52, também uma *task* para limpar os registos da tabela de factos, um *count* do número de registos iniciais e um *insert* da Chave\_Auditoria.

Então começamos por criar uma *execute SQL task* que irá primeiramente fazer um *insert* dos valores da tabela auditoria para posteriormente ligarmos um *data flow task, figura 53.*

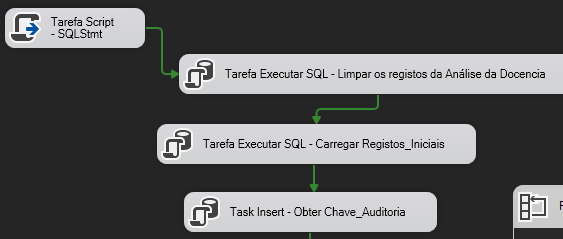


Figura – Tasks SQL

Em seguida, o processo de carregar a Staging Table, figura 53.

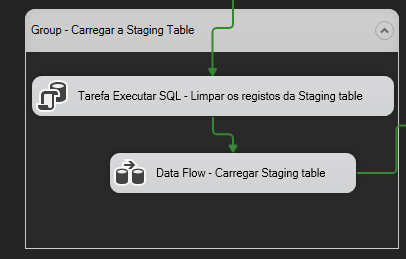


Figura – Staging Table

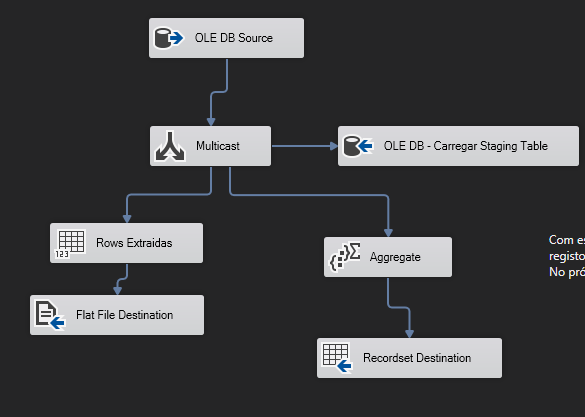


Figura – Data Flow do Carregar Staging Table

De seguida está representado na figura 54 o carregamento da staging table para juntar com a CatarinaEduardoNunoRicardo em que de seguida é divido com o multicast que faz com que seja usado um row count dos registos extraídos e com que os dados sejam agregados para um row count “destino”.

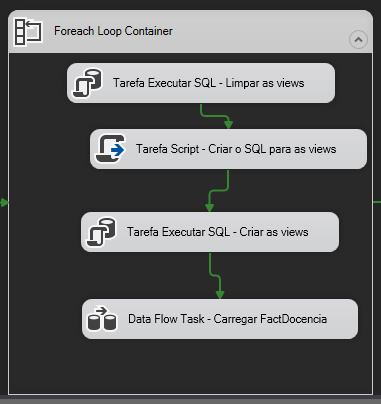


Figura – Foreach Loop Container

Este ciclo vai selecionar em todos os registos de cada data de inscrição e criar views temporárias com os registos que estavam em vigor nesses dias para cada dimensão. Depois carrega na tabela de factos final as surrogates de cada dimensão.

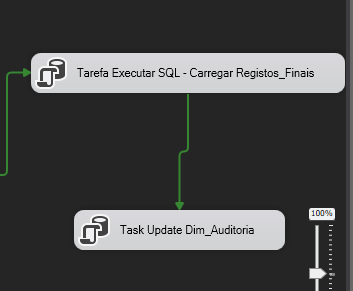


Figura -Update final da Dim\_Auditoria

De seguida é feito um carregamento dos registos finais para posteriormente ser feito um update da Dim\_Auditoria como representado na figura 56.

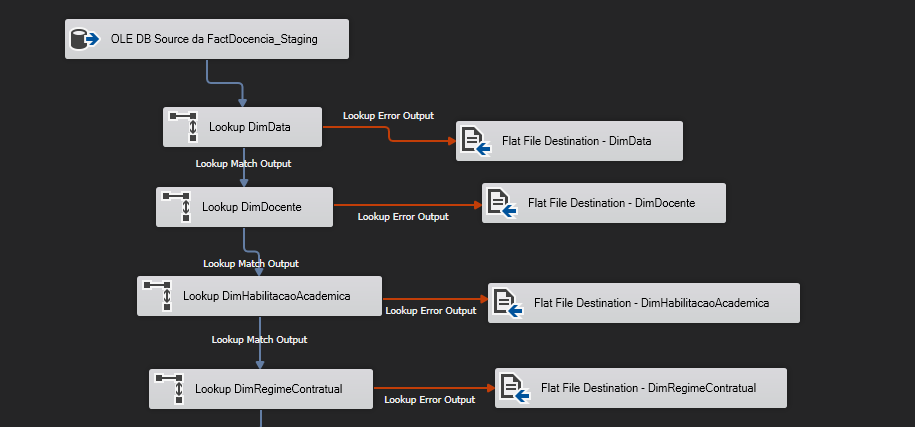


Figura 57-Data Flow Task com as verificações dos Lookups

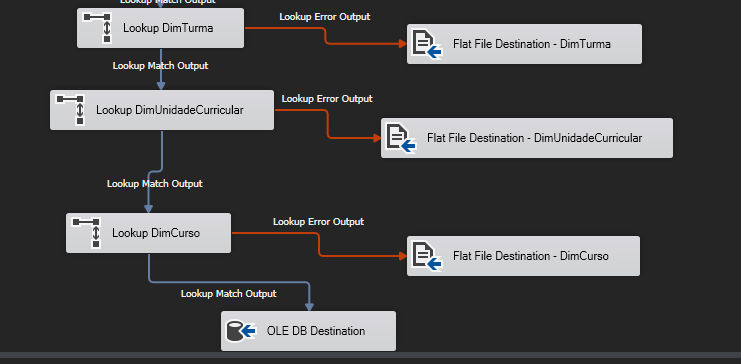


Figura - Data Flow Task com as verificações dos Lookups

Um lookup é basicamente uma tabela de consulta que habilita o desempenho de junções por igualdade simples entre a entrada e o conjunto de dados de referência, por isso criámos Lookups para cada uma das tabelas de dimensão, como está representado nas figuras 57 e 58.

Caso ocorra algum erro esse erro irá ser enviado para um flat file destination onde os dados irão ser colocados num ficheiro á parte que identificará os erros.

**Conclusão:**

# Bibliografia

Kimball, Ralph; Ross, Margy – The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling – Wiley

<https://moodle.ips.pt/2021/course/view.php?id=2225>

<https://www.oracle.com/pt/database/what-is-a-data-warehouse/>

<https://www.edureka.co/blog/dimension-table-in-data-warehousing/>

<https://www.zentut.com/data-warehouse/fact-table/>

<https://canaltech.com.br/business-intelligence/a-granularidade-de-dados-no-data-warehouse-26310/>