



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
1.1 Data Warehouse	1
1.2 Modelo de Dados	1
1.3 Modelo Dimensional.....	1
1.4 Fases do Modelo Dimensional	2
1.5 Matriz Bus	3
1.6 Matriz Prioridades	4
1.7 Processo de negócio.....	4
1.8 Granularidade	4
1.9 Descrição das dimensões	5
1.10 Star Schema	6
2. Auditoria para o Processo ETL	10
2.1 O que é o Processo ETL?	10
2.1.1 Primeira Etapa – Extração.....	11
2.1.2 Segunda Etapa – Transformação	11
2.1.3 Terceira Etapa – Carregamento	11
2.2 Tabela Auditoria	12
3. Explicação do Processo ETL	12
3.1 <i>Package</i> Início.....	12
3.2 Tabelas de Dimensão	15
3.3 Tabela de Factos	30
4. Bibliografia	36

Índice de Tabelas

Tabela 1 - Matriz Bus	3
Tabela 2 - Descrição das dimensões presentes no processo de negócio.....	5

Índice de Figuras

Figura 1 - Matriz Prioridades	4
Figura 2 - Star Schema	6
Figura 3 - Tabela de dimensão Unidade Curricular	7
Figura 4- Tabela de dimensão da Habilitação Académica	7
Figura 5 - Tabela de Dimensão do Docente	8
Figura 6 - Tabela de Dimensão do Regime Contratual	8
Figura 7 - Tabela de Dimensão da Turma	8
Figura 8- Tabela de Dimensão de Curso	9
Figura 9 - Tabela de Dimensão da Data	9
Figura 10 - Tabela de Factos da Análise da Docência	10
Figura 11 – Esquema do Processo ETL	10
Figura 12 - Tabela Auditoria com exemplos	12
Figura 13 - Package Início	12
Figura 14 - Variáveis criadas	13
Figura 15 - Inserção Inicial da Auditoria	13
Figura 16 - Configuração da Task “Inserção Inicial da Auditoria”	13
Figura 17 – Ficheiro .txt com a inserção dos values na tabela Auditoria	14
Figura 18 - Mapeamento dos parâmetros da task “Inserção Inicial da Auditoria”	14
Figura 19 - Result Set da task “Inserção Inicial da Auditoria”	15
Figura 20 - Progressão da task “Inserção Inicial da Auditoria” para a task “Execute Package DimDocente”	16
Figura 21 - Configuração da task “DimDocente”	16
Figura 22 - Relacionamento dos parâmetros da task “DimDocente”	17
Figura 23 Variáveis da task "DimDocente"	17
Figura 24 - Result Set da task "ContarRegistosIniciais"	18
Figura 25 Configuração da task “Contar Registos Iniciais”	18
Figura 26 - Progressão da task “Contar Registos Iniciais” para a task “Task Insert Auditoria Inicial”	19

Figura 27 - Ficheiro .txt com o script para fazer a inserção das variáveis iniciais na tabela auditoria.....	19
Figura 28 - Configuração da task “Task Insert Auditoria Inicial”	19
Figura 29 Configuração da task “Task Insert Auditoria Inicial”	19
Figura 30 - Progressão da task “Task Insert Auditoria Inicial” para o data-flow "PDW_DimDocente"	20
Figura 31-Ole DB Source	20
Figura 32-Colunas retiradas da tabela de funcionários	20
Figura 33 Script utilizado para retirar os valores necessários da tabela Funcionários.....	21
Figura 34-Slowly Changing Dimension	21
Figura 35 Slowly Changing Dimension.....	21
Figura 36- Contagem dos registos extraídos.....	21
Figura 37- Configuração do tipo de Coluna para cada registo	23
Figura 38- Configurações para as colunas tipo Historical	23
Figura 39 Configurações para as colunas tipo Fixed e Changing	23
Figura 40- Slowly Changing Dimension concluído com as contagens de tipo de coluna.....	24
Figura 41- Contagem das alterações de tipo.....	24
Figura 42--Contagem das alterações de tipo 1	25
Figura 43- Configuração do OLE DB Destination.....	25
Figura 44- Mapeamento das colunas de entrada com as da tabela de destino	26
Figura 45- Conexão entre o data-flow e task "Contar Registos Finais"	26
Figura 46- Configuração da task “Registos Finais”	27
Figura 47- Result Set da task “Registos Finais”	27
Figura 48- Ligação da task " Contar Registos Finais" à task "Update Auditoria"	28
Figura 49- Configuração da task "Update Auditoria"	28
Figura 50- Script em formato .txt do update da tabela auditoria.....	29
Figura 51- Variáveis que vão ser inseridas na tabela auditoria	29
Figura 52 – Tasks SQL	30
Figura 53 – Staging Table	31
Figura 54 – Data Flow do Carregar Staging Table	31

Figura 55 – Foreach Loop Container.....	32
Figura 56 – Update da Dim_Auditoria	33
Figura 57-Data Flow Task com as verificações dos Lookups	33
Figura 58- Data Flow Task com as verificações dos Lookups	34

1. Introdução

1.1 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*).

1.2 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.

1.3 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 características 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.

1.4 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 o último 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.

1.5 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 1 - Matriz Bus

<u>Processos de negócio</u>	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

1.6 Matriz Prioridades

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

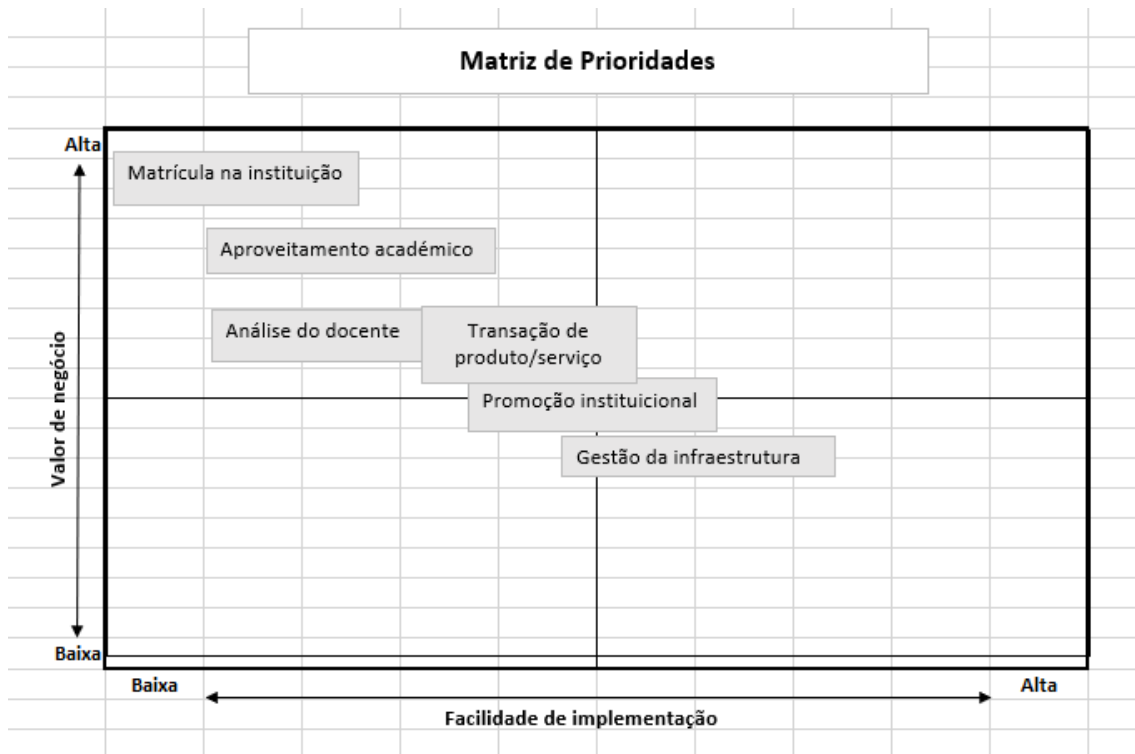


Figura 1 - Matriz Prioridades

1.7 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.

1.8 Granularidade

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

1.9 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 2 - 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 caracteriza 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.

1.10 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.

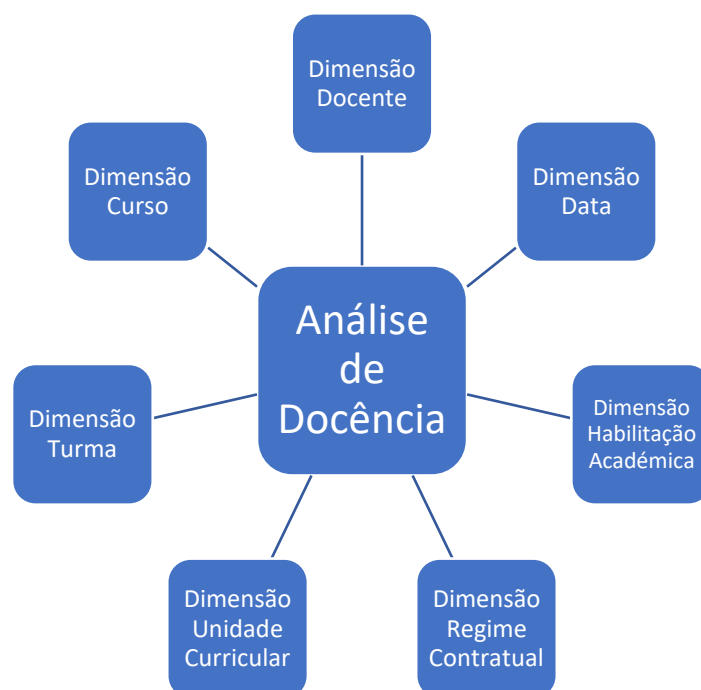


Figura 2 - Star Schema

Unidade Curricular			
PK	CodigoUnidadeCurricular	Varchar(7)	Ex:1
	ChaveUnidadeCurricular	int	Ex:1
	NomeUnidadeCurricular	Varchar(100)	Ex:Data Warehouse
	Abreviatura	Varchar(50)	Ex:DW
	AnoCurricular	Varchar(50)	Ex:2020
	ÁreaCientífica	Varchar(50)	Ex:Ciências
	Semestre	Varchar(50)	Ex:1
	Creditos	Varchar(50)	Ex:40
	Opcional	Varchar(50)	Ex:1
	Estado	Varchar(50)	Ex:Ativo
	DataInicio	smalldatetime	Ex: 2021-01-09 21:07:00
	DataFim	smalldatetime	Ex: 2021-01-09 21:07:01

Figura 3 - Tabela de dimensão Unidade Curricular

Habilitação Académica			
PK	CodigoHabilitacao	Inteiro	Ex: 1
	ChaveHabilitacao	Inteiro	Ex: 1
	DataObtenção	date	Ex:01-01-20
	CursoObtido	varchar	Ex: Bioinformática
	ÁreaDeConhecimento	varchar	Ex: Ciência
	InstituiçãoFrequentada	varchar	Ex: IPS
	ClassificaçãoObtida	double	Ex: 18
	TítuloDaTese	varchar	Ex: "Análises de dados"

Figura 4- Tabela de dimensão da Habilitação Académica

Docente			
PK	CodigoDocente	Inteiro	Ex: 1
	ChaveDocente		Ex: 1
	Nome	Varchar(80)	Ex: Nuno
	DataNascimento	Varchar(50)	Ex: 29-01-1998
	Localidade	Varchar(50)	Ex: Alverca
	Concelho	Varchar(100)	Ex: Vila Franca de Xira
	Distrito	Varchar(100)	Ex: Lisboa
	Nacionalidade	Varchar(50)	Ex: Portuguesa
	Numeroidentificação	Varchar(20)	Ex: 12345678
	TipolD	Varchar(50)	Ex: CC
	EstadoCivil	Varchar(50)	Ex: Solteiro
	Dependentes	Varchar(50)	Ex: 1
	Sexo	Varchar(50)	Ex: M
	CódigoGabinete	Varchar(50)	Ex: 1234
	ExtensãoTelefónica	Varchar(50)	Ex: 1234
	DataInicio	smalldatetime	Ex: 2021-01-09 21:07:00
	DataFim	smalldatetime	Ex: 2021-01-09 21:07:01

Figura 5 - Tabela de Dimensão do Docente

Regime Contratual			
PK	CodigoRegime	Inteiro	Ex:1
	ChaveRegime	int	Ex:1
	Designação	Varchar(50)	Ex:21
	HorasContrato	Varchar(50)	Ex:100
	HorasLetivas	Varchar(50)	Ex:100
	ETITotal	Varchar(50)	Ex:100
	DataInicio	smalldatetime	Ex: 2021-01-09 21:07:01
	DataFim	smalldatetime	Ex: 2021-01-09 21:07:02

Figura 6 - Tabela de Dimensão do Regime Contratual

Turma			
PK	CodigoTurma	Inteiro	Ex:1
	ChaveTurma	int	Ex:1
	DesignacaoDaTurma	Varchar(50)	Ex:21
	MáximoNovos	Varchar(50)	Ex:100
	MáximoRepetentes	Varchar(50)	Ex:50
	DataInicio	smalldatetime	Ex: 2021-01-09 21:07:00
	DataFim	smalldatetime	Ex: 2021-01-09 21:07:01

Figura 7 - Tabela de Dimensão da Turma

Curso			
PK	CodigoCurso	Varchar(50)	Ex:1
	ChaveCurso	int	Ex:1
	Ciclo	Varchar(50)	Ex:1
	Regime	Varchar(50)	Ex:misto
	CódigoRegime	Varchar(50)	Ex:1
	NomeCurso	Varchar(50)	Ex:Bioinformática
	Abreviatura	Varchar(50)	Ex:Bioinf
	Acronimo	Varchar(50)	Ex:bioinf
	GrauAcadémico	Varchar(50)	Ex:mestrado
	Estado	Varchar(50)	Ex:Ativo
	PrincipaisAreasEstudoEN	Varchar(50)	Ex:Inglês
	DesignaçãoEstatutoInstituição	Varchar(50)	Ex:1
	LinguasAprendizagem	Varchar(50)	Ex:Francês
	NivelQualificação	Varchar(50)	Ex:20
	DuraçãoOficial	Varchar(50)	Ex:1200
	SistemaQualificação	Varchar(50)	Ex:ECTS
	ECTS	Varchar(50)	Ex:100
	DataInicio	smalldatetime	Ex: 2021-01-09 21:07:00
	DataFim	smalldatetime	Ex: 2021-01-09 21:07:01

Figura 8- Tabela de Dimensão de Curso

Data			
PK	CodigoData	Inteiro	Ex:1
	ChaveData	int	Ex: 1
	DataSQL	date	Ex: 20
	DescriçãoData	Varchar(50)	Ex: 20
	DiaDaSemana	Varchar(50)	Ex: quarta
	MêsDoAno	Varchar(50)	Ex: janeiro
	DiaDoAno	Varchar(50)	Ex: 12
	DiaDoMês	Varchar(50)	Ex: 12
	NúmeroDaSemana	Varchar(50)	Ex: 1
	IndicadorFeriado	Varchar(50)	Ex: F
	IndicadorFimDeSemana	Varchar(50)	Ex: FDS
	SemestreEscolar	Varchar(50)	Ex: 1
	SemestreEfetivo	Varchar(50)	Ex: 1
	AnoEscolar	Varchar(50)	Ex: 2020/2021
	AnoCivil	Varchar(50)	Ex: 2020

Figura 9 - Tabela de Dimensão da Data

Análise da Docência			
PK	CodigoDocencia	Inteiro	Ex:1
	ChaveDocente	int	Ex:1
	ChaveHabilitacao	int	Ex:1
	ChaveRegime	int	Ex:1
	ChaveCurso	int	Ex:1
	ChaveDisciplina	int	Ex:1
	ChaveTurma	int	Ex:1
	ChaveData	int	Ex:1
	NumHorasLecionadas	int	Ex:1
	NumHorasRegimeNoturno	int	Ex:1

Figura 10 - Tabela de Factos da Análise da Docência

2. Auditoria para o Processo ETL

2.1 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.

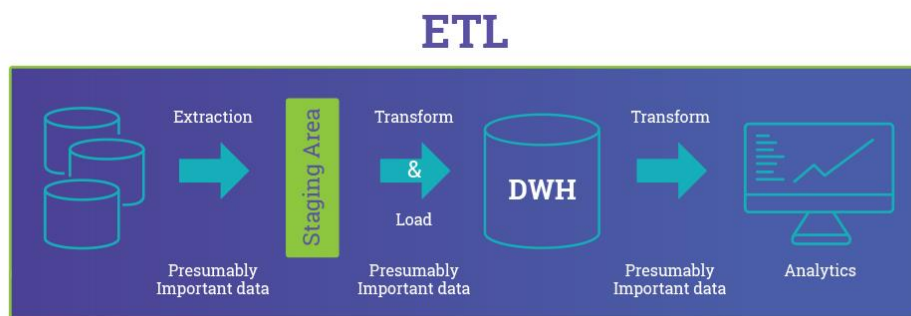


Figura 11 – Esquema do Processo ETL

2.1.1 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*.

2.1.2 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.

2.1.3 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

2.2 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.

Chave_Auditoria	Chave_AuditPAI	Nome_Package	GUID_Package	Inicio_Processo	Fim_Processo	Num_RegExtraídos	Num_RegInseridos	Registos_Iniciais	Registos_Finais	Registos_Erro	Alt_Tipo1	Alt_Tipo2	Nome_Utilizador
479	479	-1	Pkg_Inicio	{C82E9523-7...	2021-01-16 00:00:56.000	NULL	NULL	NULL	NULL	NULL	NULL	NULL	DESKTOP-KP...
480	480	479	DimDocente	{84359EEE-1...	2021-01-16 00:00:58.000	2021-01-16 ...	74	74	74	0	0	0	DESKTOP-KP...
481	481	479	DimRegimeCo...	{57FBA5E5-7...	2021-01-16 00:01:05.000	2021-01-16 ...	19	0	19	19	0	0	DESKTOP-KP...
482	482	479	DimCurso	{8EEADCD1-...	2021-01-16 00:01:11.000	2021-01-16 ...	30	0	30	30	0	0	DESKTOP-KP...
483	483	479	DimUnidadeC...	{772CC460-6...	2021-01-16 00:01:17.000	2021-01-16 ...	811	0	4055	4055	0	0	DESKTOP-KP...
484	484	479	DimTurma	{727DEC28-C...	2021-01-16 00:01:37.000	2021-01-16 ...	895	603	3260	3863	0	0	DESKTOP-KP...
485	485	479	DimHabilitaca...	{AE4978FB-E...	2021-01-16 00:02:17.000	2021-01-16 ...	54	0	702	702	0	0	DESKTOP-KP...
486	486	479	DimData	{33D9CC64-6...	2021-01-16 00:02:21.000	2021-01-16 ...	3344	3344	0	3344	0	0	DESKTOP-KP...

Figura 12 - Tabela Auditoria com exemplos

3. Explicação do Processo ETL

3.1 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.

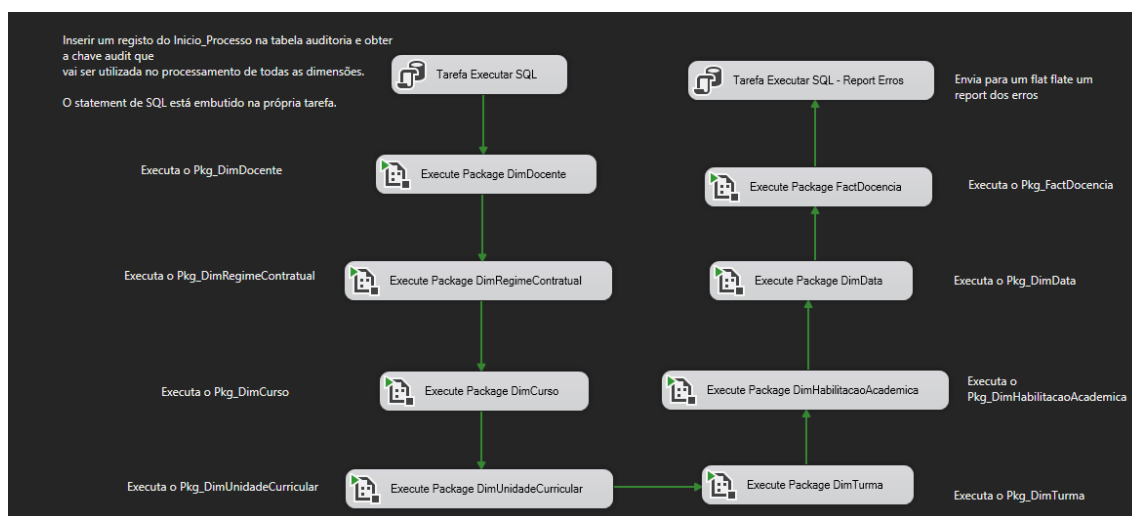


Figura 13 - Package Início


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.

Name	Scope	Data type	Value
Chave_Auditoria	Package1	Int32	0
Chave_AuditPAI	Package1	Int32	0
Fim_Processo	Package1	DateTime	08/01/2021 10:17
GUID_Package	Package1	String	
Inicio_Processo	Package1	DateTime	08/01/2021 10:18
Nome_Package	Package1	String	Pkg_Inicio

Figura 14 - Variáveis criadas

Inserir um registo do Inicio_Processo na tabela auditoria e obter a chave audit que vai ser utilizada no processamento de todas as dimensões.

O statement de SQL está embutido na própria tarefa.

 Inserção Inicial da Auditoria




Figura 15 - 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*.

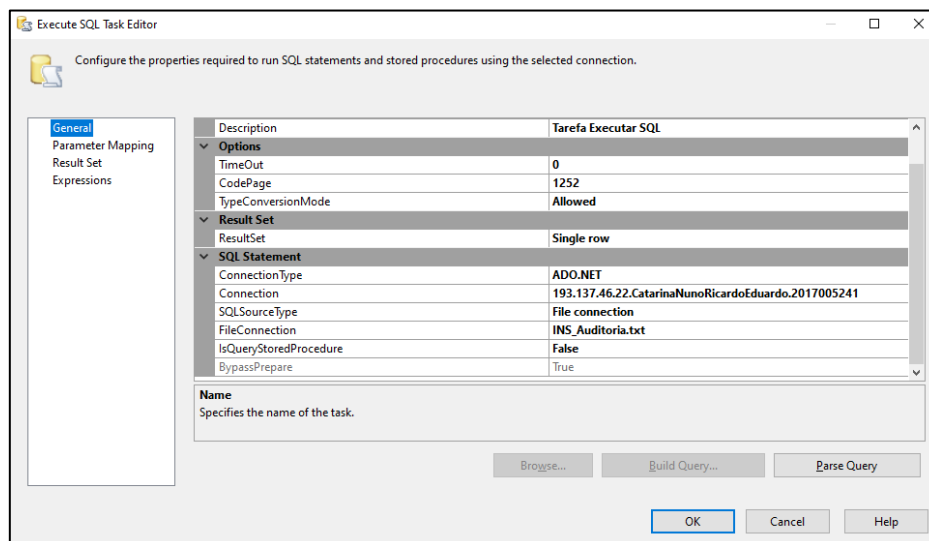


Figura 16 - Configuração da Task “Inserção Inicial da Auditoria”

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.

```

INS_Auditoria - Bloco de notas
Ficheiro Editar Formatar Ver Ajuda
INSERT INTO [PDW].[Auditoria](Chave_AuditPAI,Nome_Package,GUID_Package,Nome_Utilizador,Inicio_Processo) VALUES (-1,@Nome_Package,@GUID_Package,@Nome_Utilizador,@Inicio_Processo)
SELECT cast(IDENT_CURRENT( '[PDW].[Auditoria]') AS int) AS Chave_Auditoria

```

Figura 17 – 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.

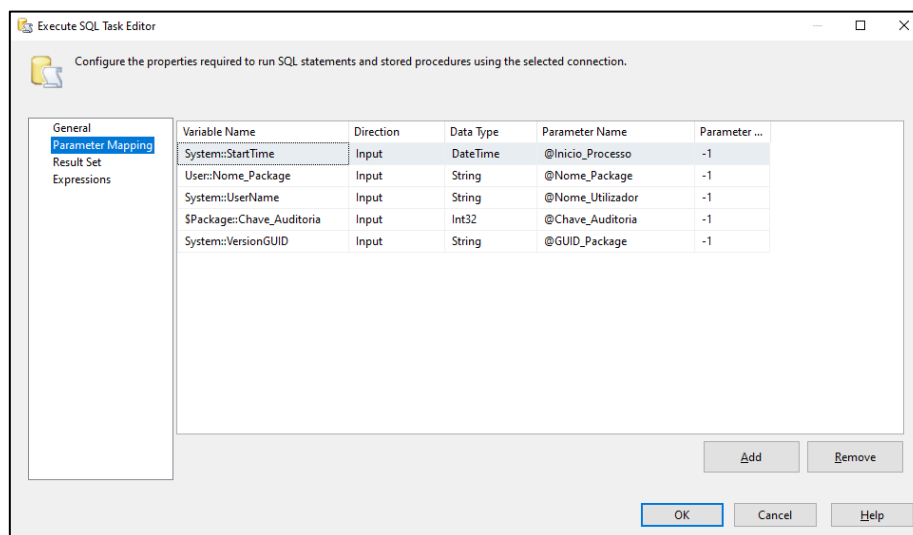


Figura 18 - Mapeamento dos parâmetros da task “Inserção Inicial da Auditoria”

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.

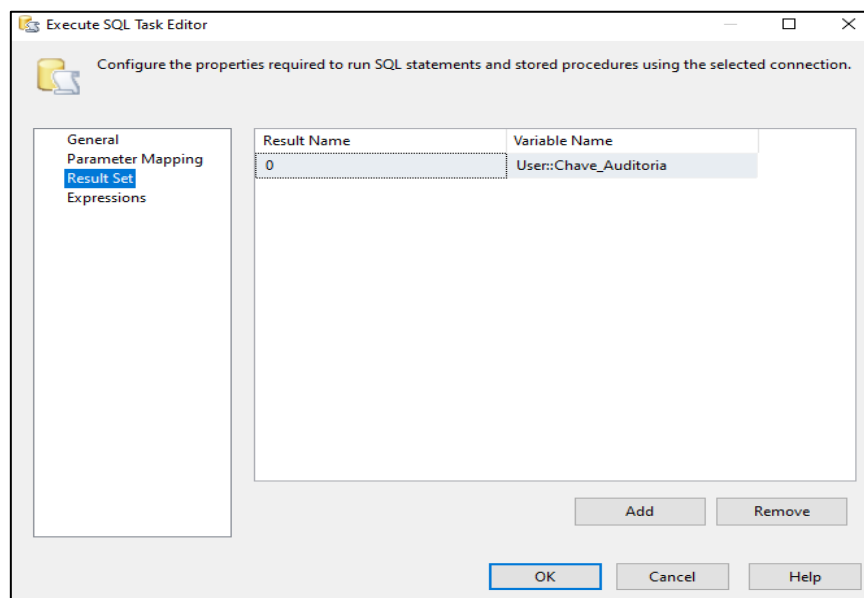


Figura 19 - Result Set da task “Inserção Inicial da Auditoria”

3.2 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.

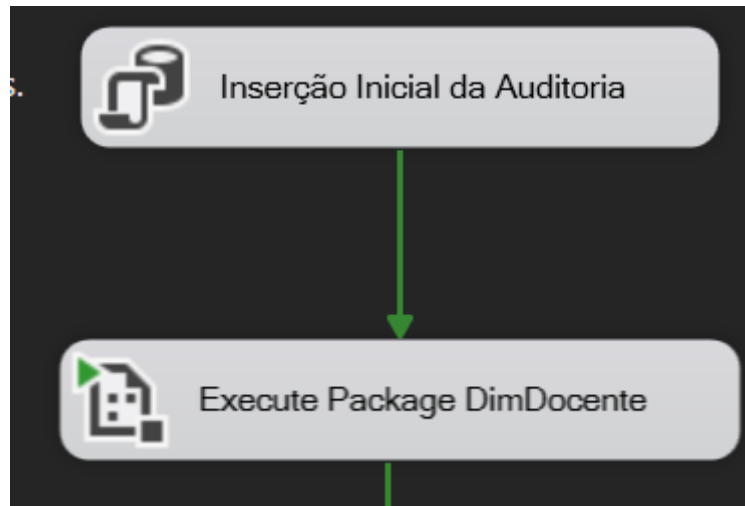


Figura 20 - Progressão da task “Inserção Inicial da Auditoria” para a task “Execute Package DimDocente”

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.

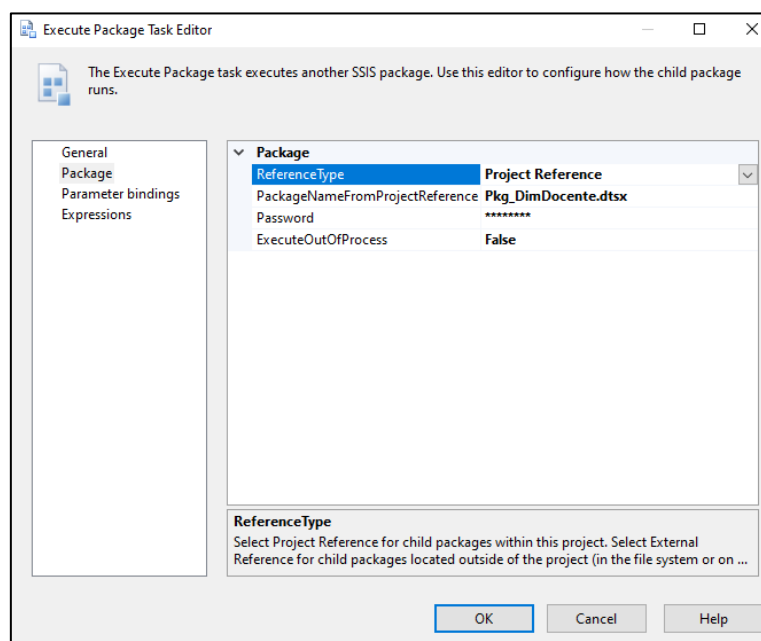


Figura 21 - Configuração da task “DimDocente”

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.

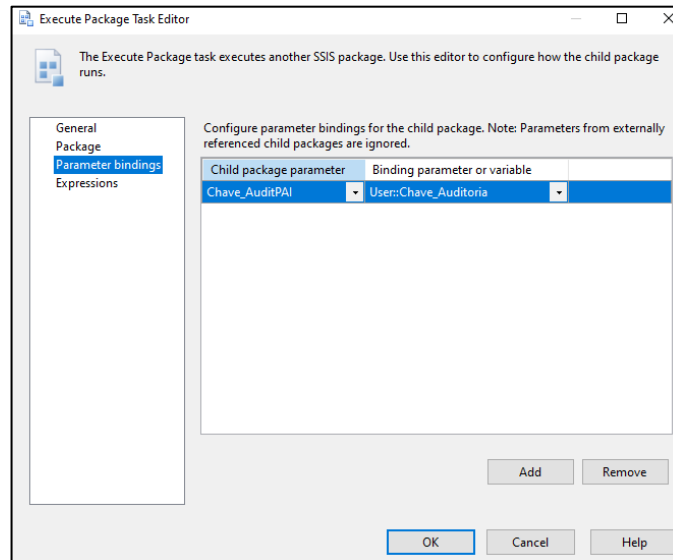


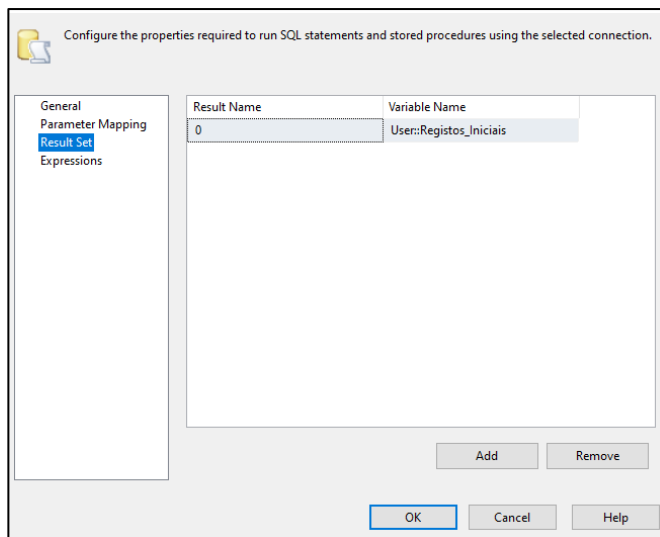
Figura 22 - Relacionamento dos parâmetros da task “DimDocente”

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

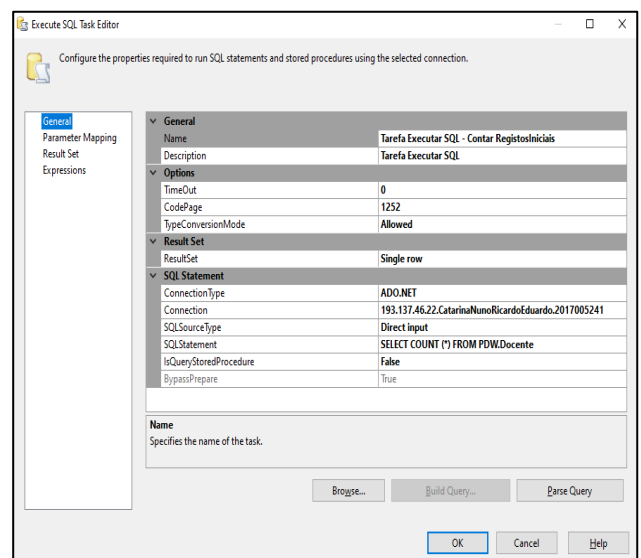
Name	Scope	Data type	Value	Expression
Alt_Tipo1	Pkg_DimDoce...	Int32	0	
Alt_Tipo2	Pkg_DimDoce...	Int32	0	
Chave_Auditoria	Pkg_DimDoce...	Int32	0	
Chave_AuditPAI	Pkg_DimDoce...	Int32	0	
Fim_Processo	Pkg_DimDoce...	DateTime	02/01/2021 16:43	
GUID_Package	Pkg_DimDoce...	Int32	0	
Inicio_Processo	Pkg_DimDoce...	DateTime	02/01/2021 16:44	
Nome_Package	Pkg_DimDoce...	String	DimDocente	
Num_RegExtrados	Pkg_DimDoce...	Int32	0	
Num_RegInseridos	Pkg_DimDoce...	Int32	0	
Registos_Erro	Pkg_DimDoce...	Int32	0	
Registos_Finais	Pkg_DimDoce...	Int32	0	
Registos_Iniciais	Pkg_DimDoce...	Int32	0	

Figura 23 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 25, sendo este valor colocado numa variável num *result set*, referencia á figura 24.



**Figura 25 - Result Set da task
"ContarRegistosIniciais"**



**Figura 24 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.

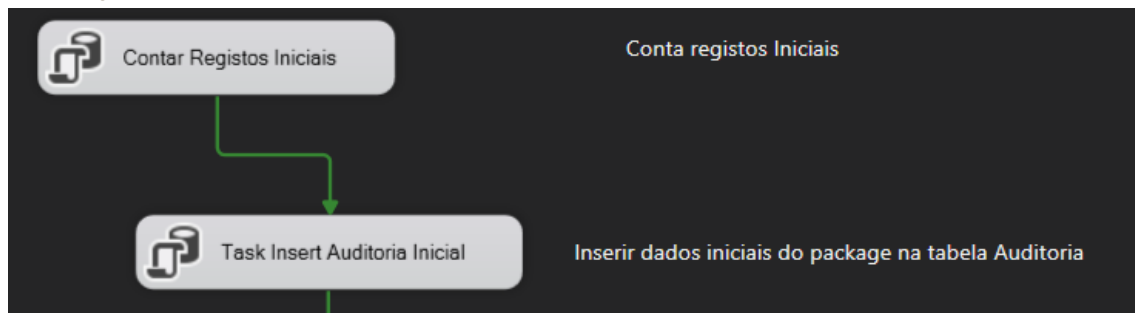


Figura 26 - Progressão da task “Contar Registos Iniciais” para a task “Task Insert Auditoria Inicial”

```

INS_PackagesAudit - Bloco de notas
Ficheiro Editar Formatar Ver Ajuda
[INSERT INTO [PDW].[Auditoria]([Nome_Utilizador],[Inicio_Processo],[Nome_Package] ,GUID_Package,[Chave_AuditPAI])
VALUES (@Nome_Utilizador,@Inicio_Processo,@Nome_Package,@GUID_Package,@Chave_AuditPAI)

SELECT cast(IDENT_CURRENT( '[PDW].[Auditoria]') AS int) AS Chave_Auditoria
  
```

Figura 27 - Ficheiro .txt com o script para fazer a inserção das variáveis iniciais na tabela auditoria

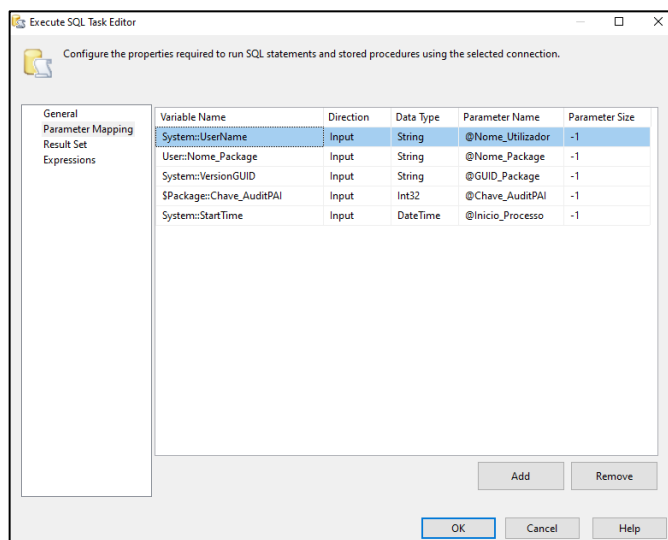


Figura 29 Configuração da task “Task Insert Auditoria Inicial”

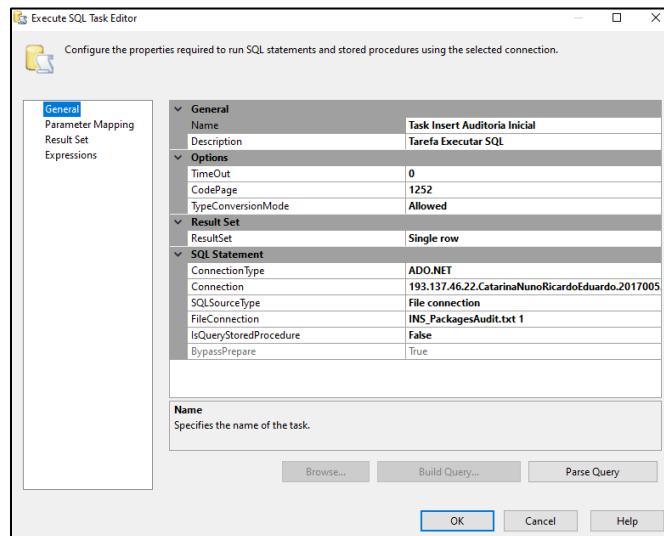


Figura 28 - Configuração da task “Task Insert Auditoria Inicial”

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

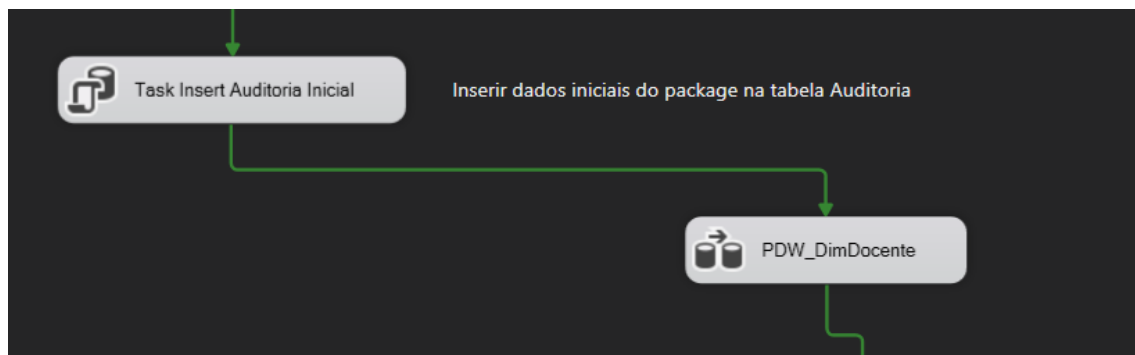


Figura 30 - Progressão da task “Task Insert Auditoria Inicial” para o data-flow “PDW_DimDocente”

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.

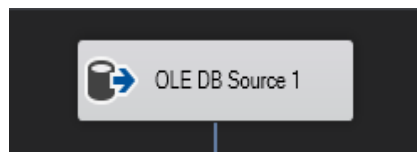


Figura 31 - OLE DB Source

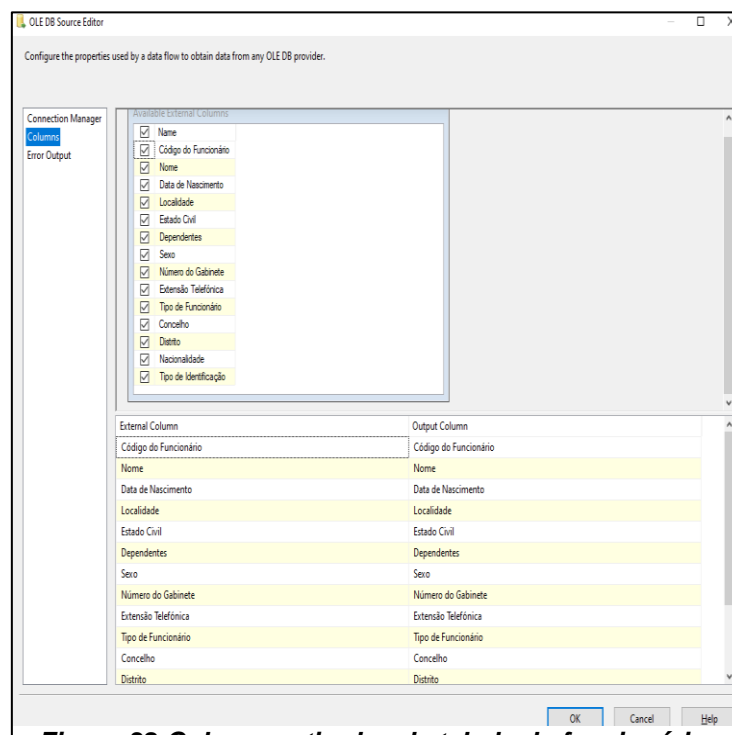


Figura 32-Colunas retiradas da tabela de funcionários

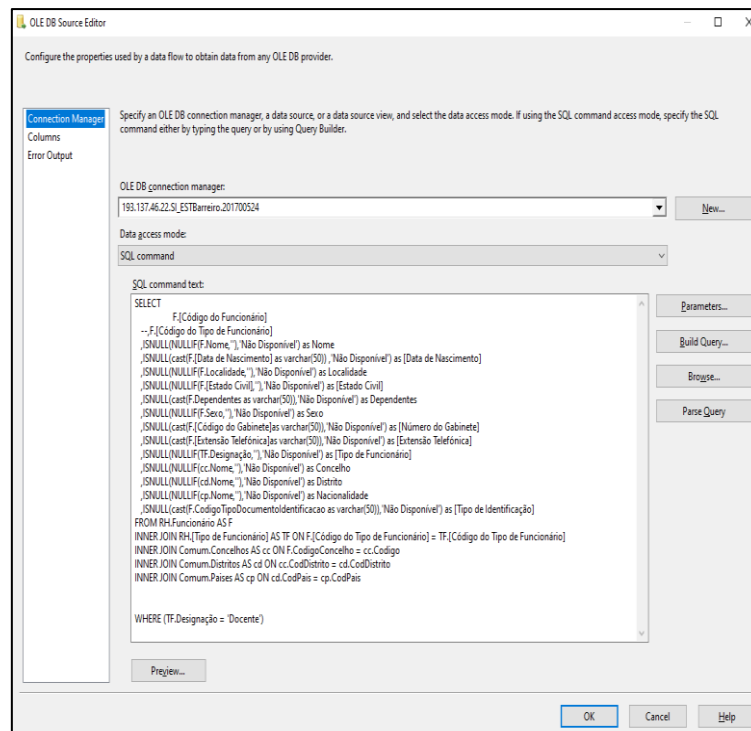


Figura 33 Script utilizado para retirar os valores necessários da tabela Funcionários

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.

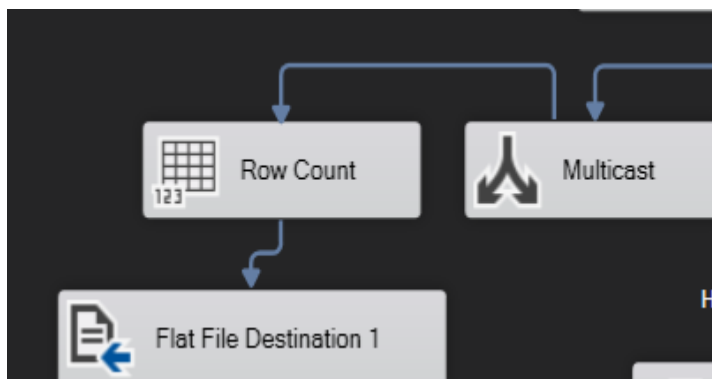


Figura 36- Contagem dos registos extraídos

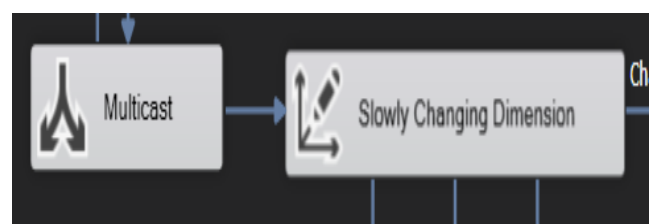


Figura 34-Slowly Changing Dimension

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.

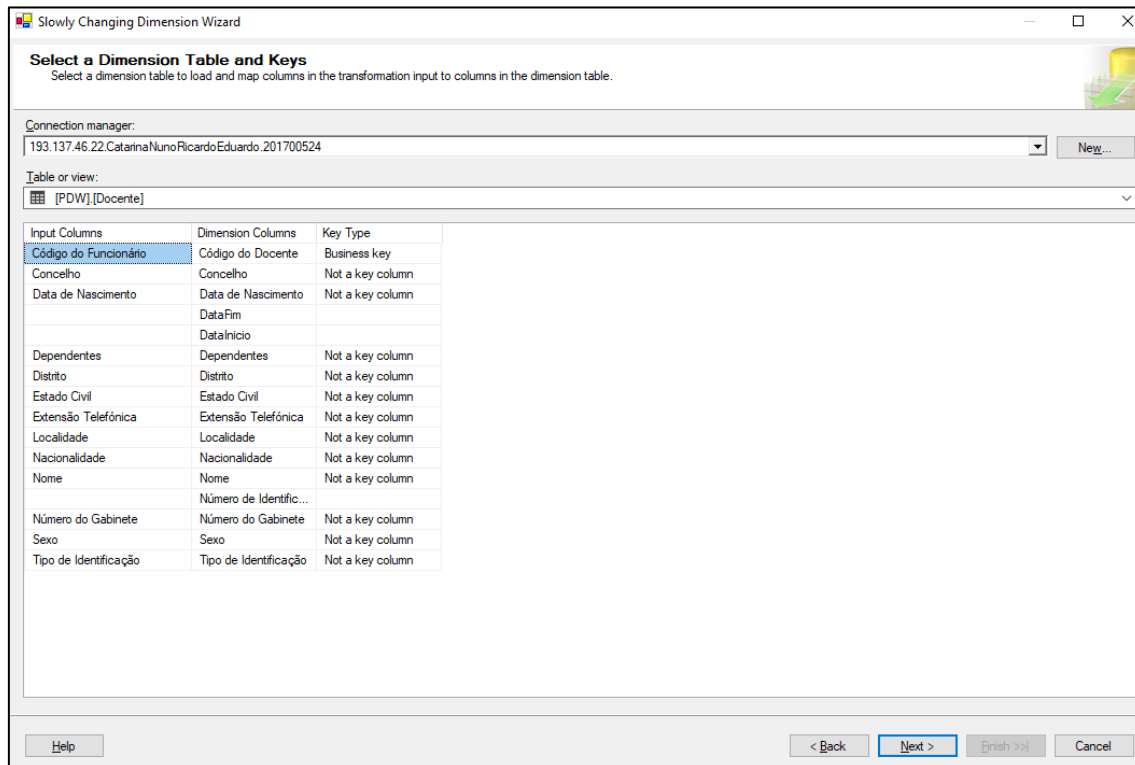


Figura 35- Atribuição da business key na Slowly Changing Dimension

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.

Slowly Changing Dimension Columns
Manage the changes to column data in your slowly changing dimensions by setting the change type for dimension columns.

Fixed Attribute
Select this type when the value in a column should not change. Changes are treated as errors.

Changing Attribute
Select this type when changed values should overwrite existing values. This is a Type 1 change.

Historical Attribute
Select this type when changes in column values are saved in new records. Previous values are saved in records marked as outdated. This is a Type 2 change.

Select a change type for slowly changing dimension columns:

Dimension Columns	Change Type
Concelho	Historical attribute
Data de Nascimento	Fixed attribute
Dependentes	Changing attribute
Distrito	Historical attribute
Estado Civil	Changing attribute
Extensão Telefónica	Changing attribute
Localidade	Historical attribute
Nacionalidade	Fixed attribute
Nome	Historical attribute
Número do Gabinete	Historical attribute
Sexo	Fixed attribute
Tipo de Identificação	Changing attribute

Remove

Help < Back Next > Finish >> Cancel

Figura 37- 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.

Historical Attribute Options
You can record historical attributes using a single column or start and end date columns.

☐ Use a single column to show current and expired records
Column to indicate current record: [dropdown]
Value when current: [dropdown]
Expiration value: [dropdown]

☒ Use start and end dates to identify current and expired records
Start date column: DataInicio [dropdown]
End date column: DataFim [dropdown]
Variable to set date values: System::CreationDate [dropdown]

Help < Back Next > Finish >> Cancel

Figura 38- Configurações para as colunas tipo Historical

Fixed and Changing Attribute Options

Fixed attributes
☒ Fail the transformation if changes are detected in a fixed attribute

Changing attributes
☐ Change all the matching records, including outdated records, when changes are detected in a changing attribute

Help < Back Next > Finish >> Cancel

Figura 39 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.

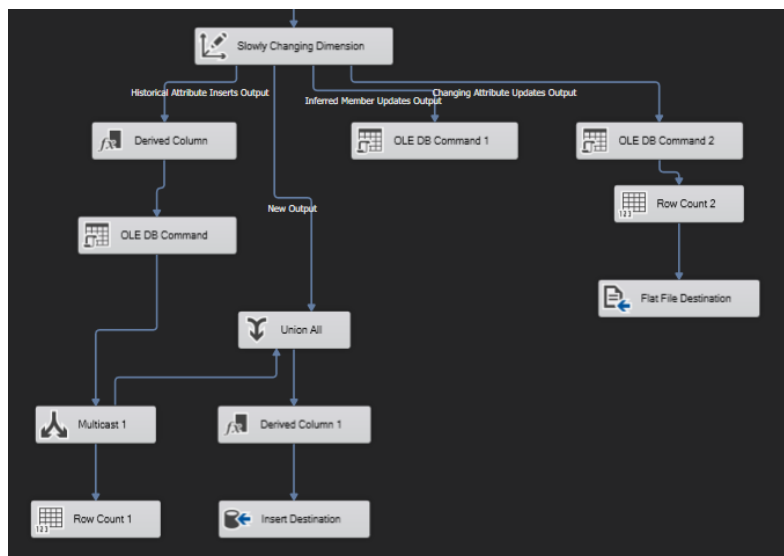


Figura 40- Slowly Changing Dimension concluído com as contagens de tipo de coluna

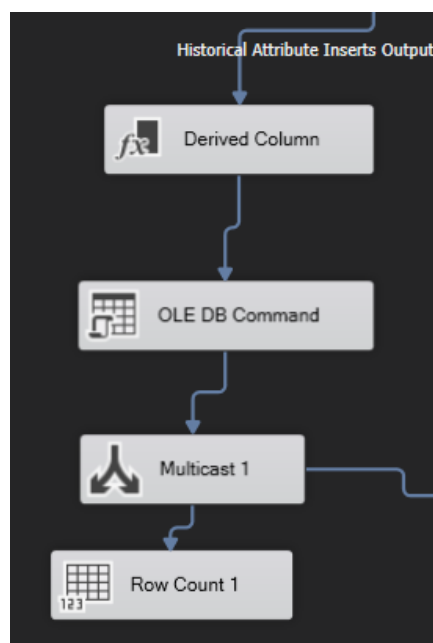


Figura 41- Contagem das alterações de tipo

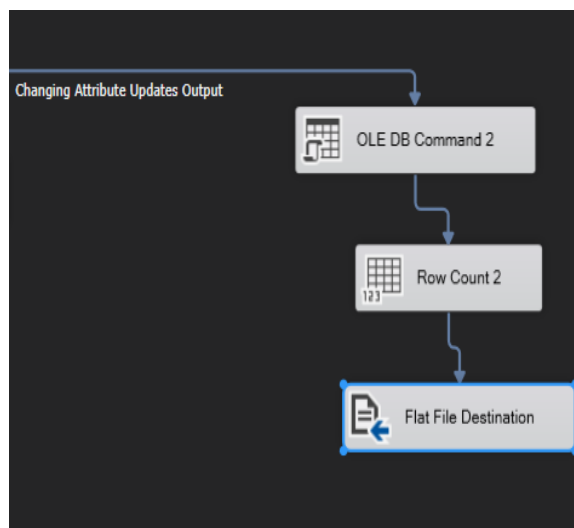


Figura 42--Contagem das alterações de tipo
1

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.

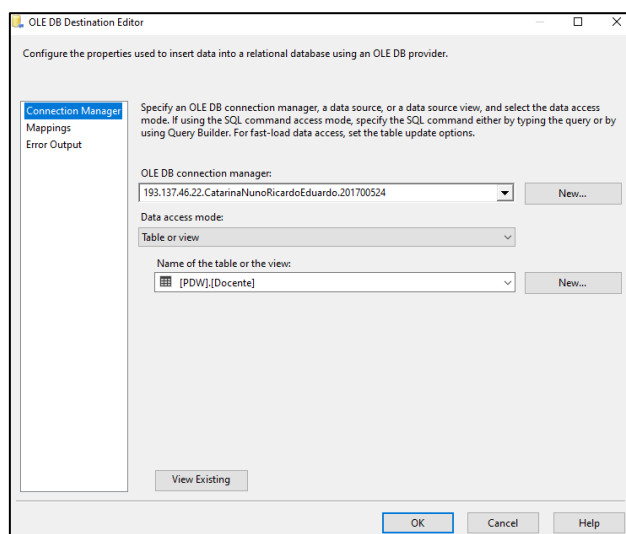


Figura 43- Configuração do OLE DB Destination

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

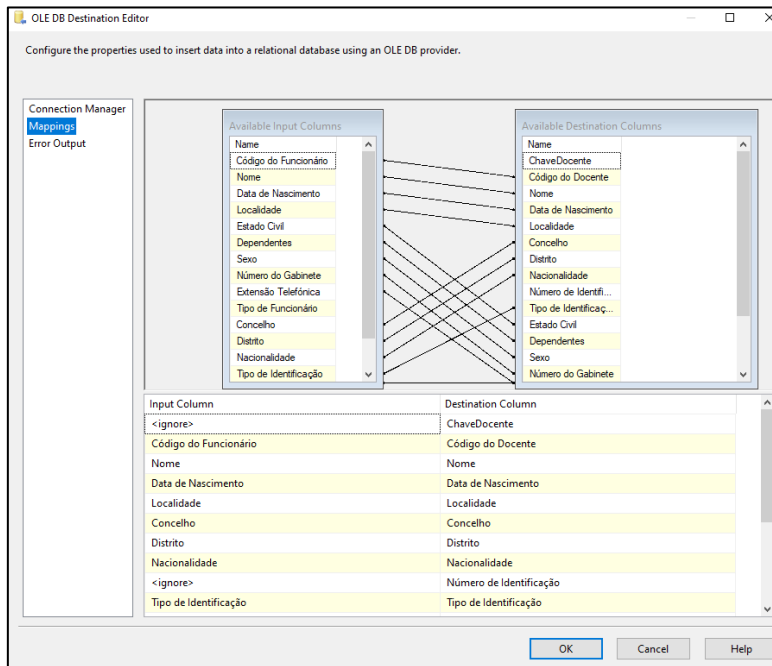


Figura 44- Mapeamento das colunas de entrada com as da tabela de destino

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.

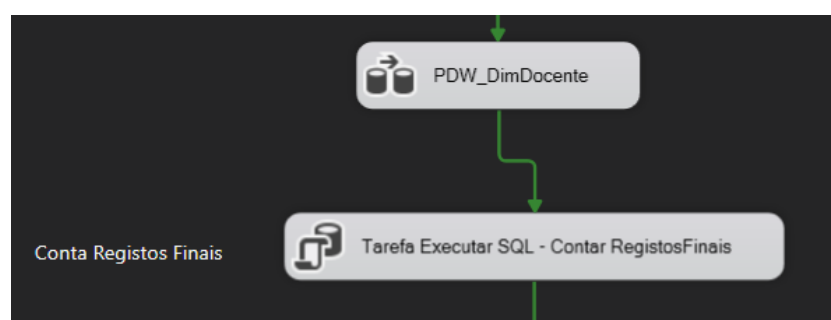


Figura 45- Conexão entre o data-flow e task "Contar Registos Finais"

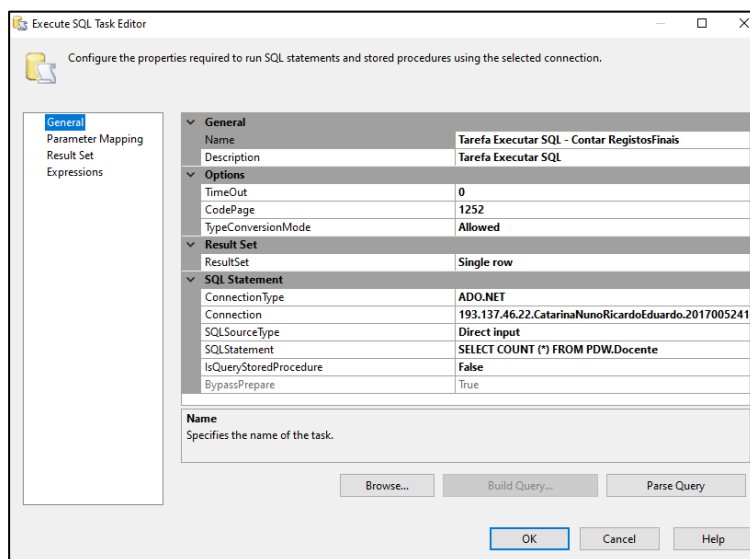


Figura 46- Configuração da task “Registos Finais”

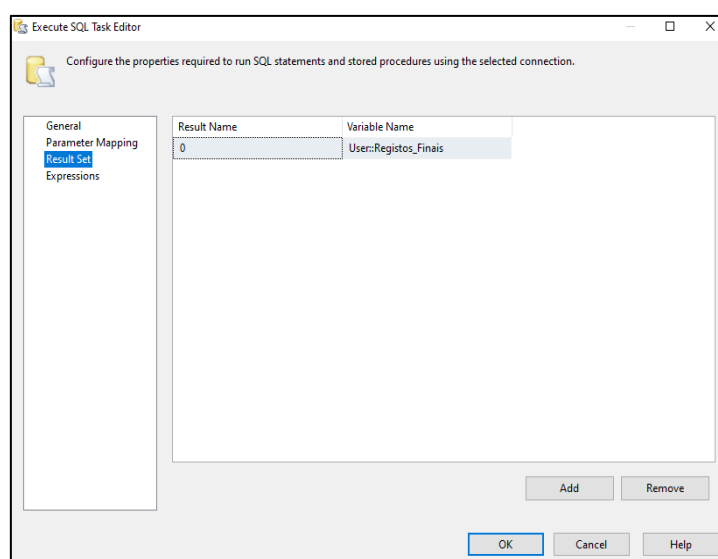


Figura 47- Result Set da task “Registos Finais”

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

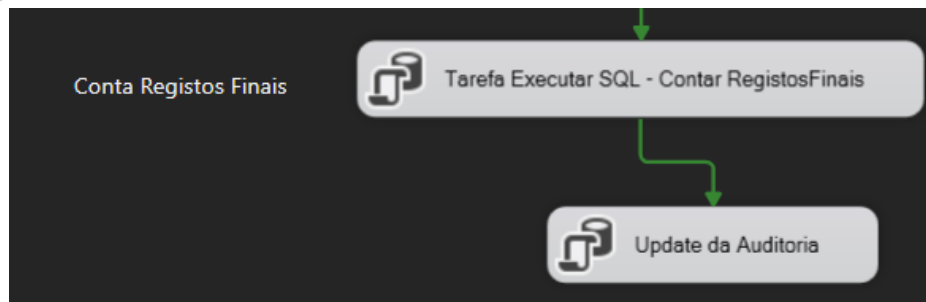


Figura 48- Ligação da task " Contar Registos Finais" à task "Update Auditoria"

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.

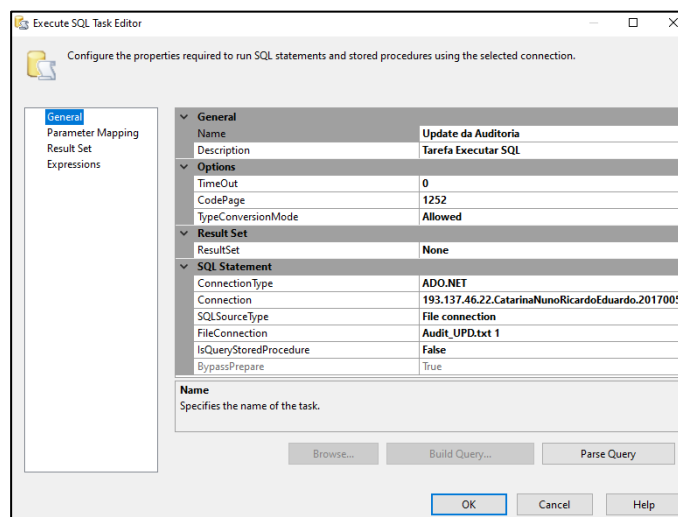


Figura 49- Configuração da task "Update Auditoria"

```
Audit_UPD - Bloco de notas
Ficheiro  Editar  Formatar  Ver  Ajuda
UPDATE PDW.Auditoria SET

Fim_Processo = getdate(),
Nome_Package = @Nome_Package,
Registos_Iniciais = @Registos_Iniciais,
Registos_Finais = @Registos_Finais,
Num_RegInseridos = @Registos_Finais - @Registos_Iniciais,
Num_RegExtraidos = @Num_RegExtraidos,
Registos_Erro = @Registos_Erro,
Alt_Tipo1 = @Alt_Tipo1,
Alt_Tipo2 = @Alt_Tipo2

WHERE Chave_Auditoria = @Chave_Auditoria
```

Figura 50- Script em formato .txt do update da tabela auditoria

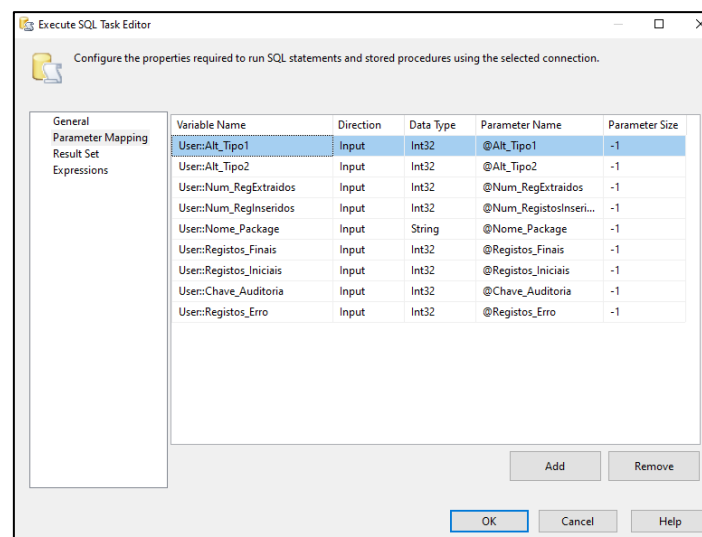


Figura 51- Variáveis que vão ser inseridas na tabela auditoria

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

3.3 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*.

Começamos 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.

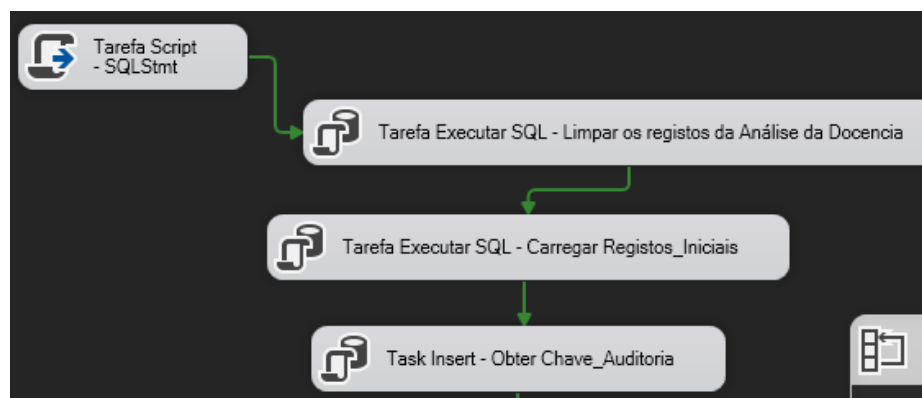


Figura 52 – Tasks SQL

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

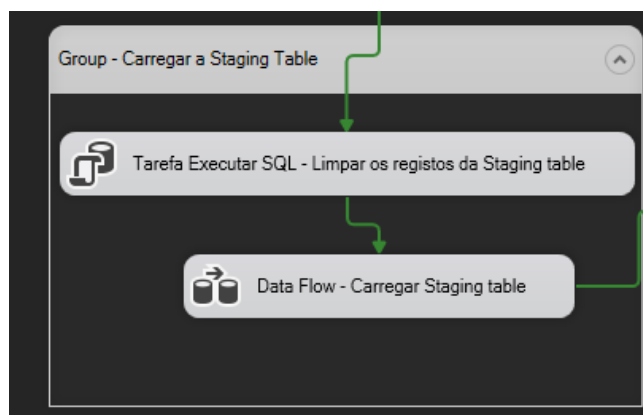


Figura 53 – Staging Table

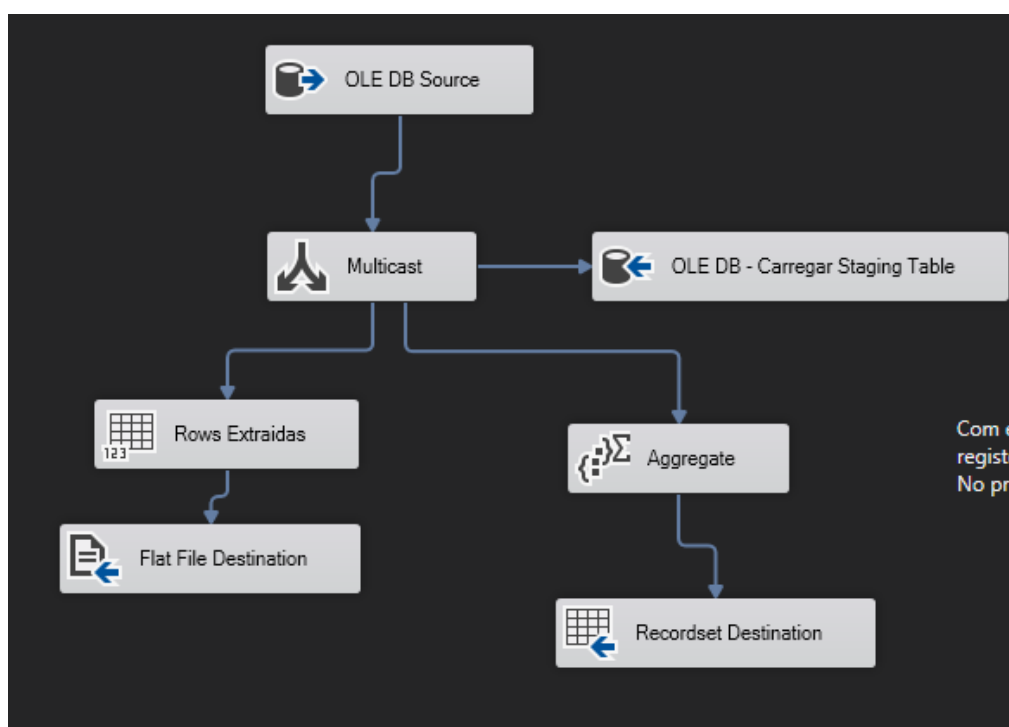


Figura 54 – Data Flow do Carregar Staging Table

De seguida está representado na figura 54 o carregamento da *staging table* para juntar com a DB CatarinaNunoRicardoEduardo, em que de seguida é dividido 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”.

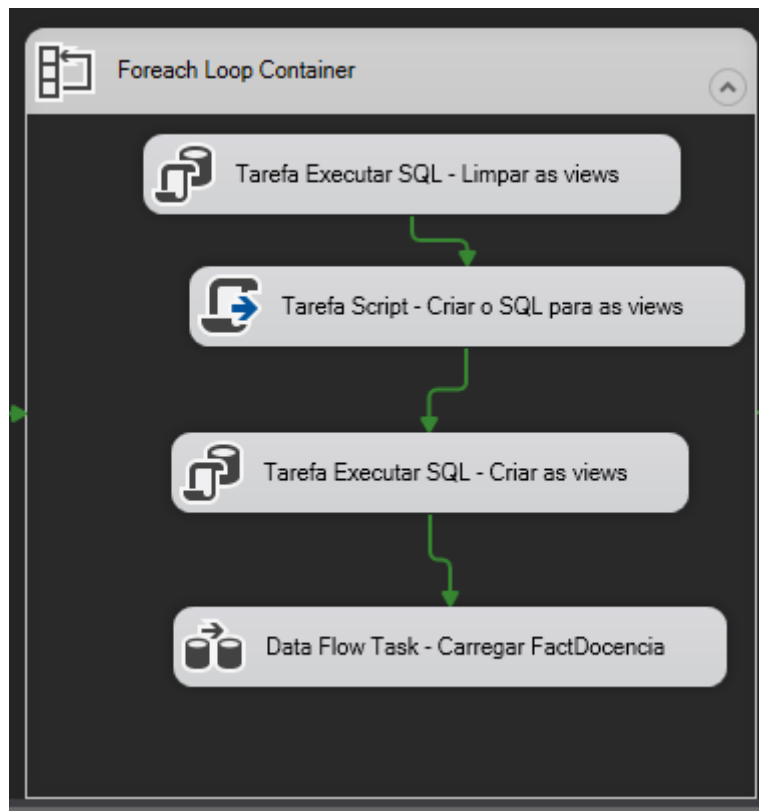


Figura 55 – Foreach Loop Container

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.

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

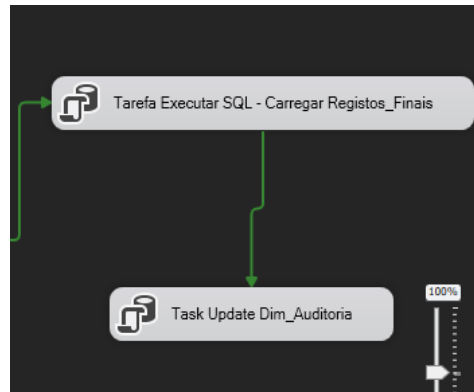


Figura 56 – Update da Dim Auditoria

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.

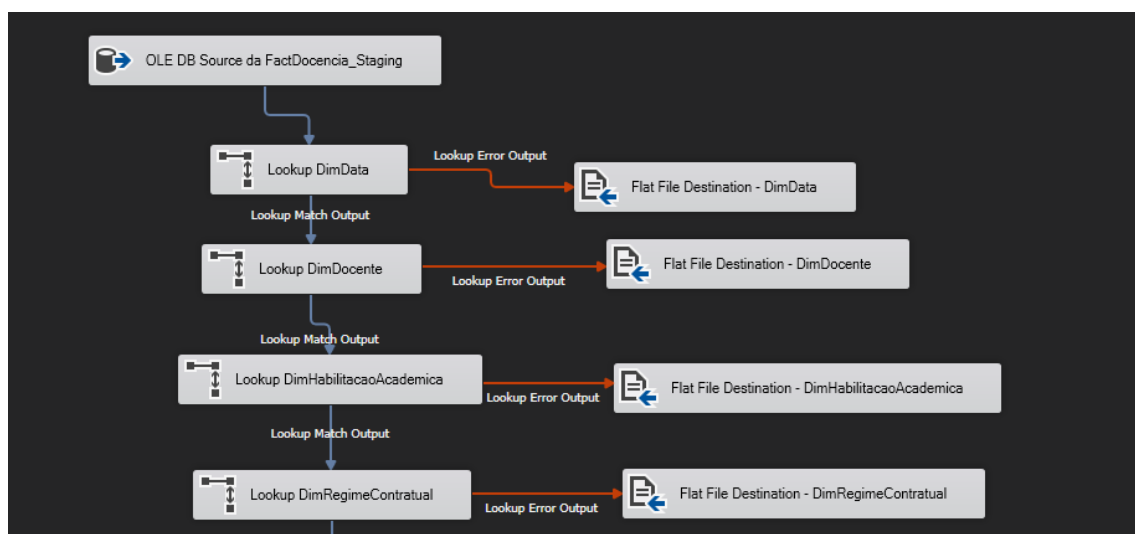


Figura 57-Data Flow Task com as verificações dos Lookups

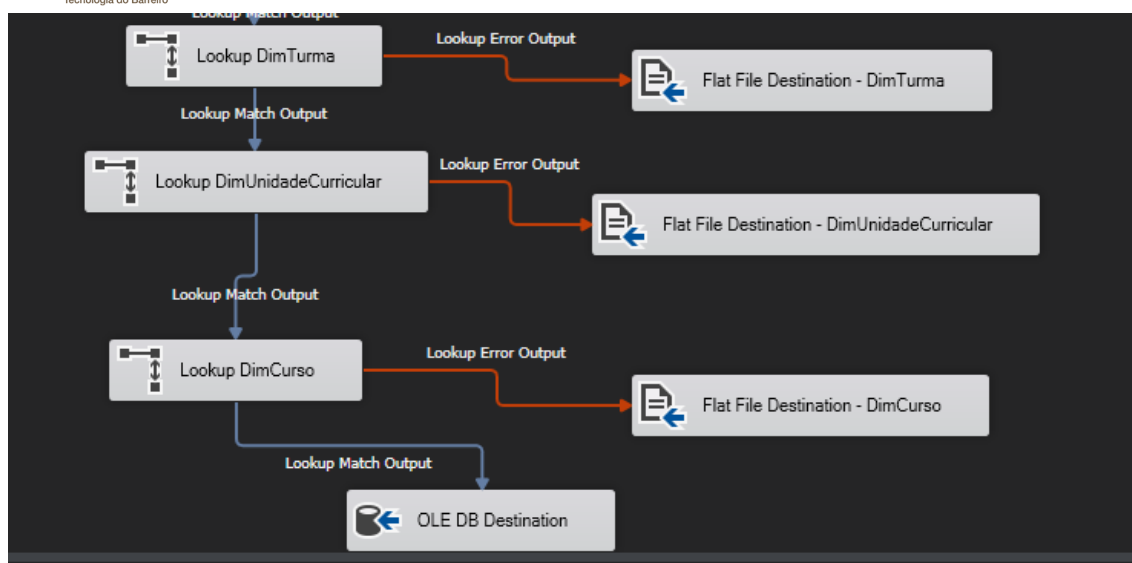


Figura 58- Data Flow Task com as verificações dos Lookups

Conclusão

Com este projeto conseguimos aperfeiçoar e melhorar o nosso conhecimento em *Data Warehouse*, que futuramente serão essenciais no mercado de trabalho.

Neste projeto utilizámos dois *softwares* utilizados como ambiente de trabalho (*Visual Studio* (SSIS) e *SQL Management Studio* (SSMS)) e também utilizámos alguns ficheiros de apoio em formato *Excel*. Conseguimos superar as diversas dificuldades que surgiram ao longo da realização do trabalho.

Concluindo, através da implementação e modelação dimensional do processo de negócio e criação do processo ETL do mesmo, percebemos melhor como se desenvolve esta área.

4. 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/>