

INSTITUTO POLITÉCNICO DE VIANA DO CASTELO

ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

ENGENHARIA INFORMÁTICA

ADMINISTRAÇÃO DE BASES DE DADOS

2023/2024

Trabalho Prático

Alunos:

Pedro Correia - 29241 Pedro

Poças - 28976 Afonso

Fernandes - 29344 Tiago

Baptista - 28998

Docente:

Marco Lima



**Instituto Politécnico
de Viana do Castelo**

2 de junho de 2024

Conteúdo

1	INTRODUÇÃO	1
2	FILESTREAM E PROCEDIMENTOS DE INSERÇÃO	3
2.1	HABILITAR FILESTREAM NO SSCM	3
2.2	CRIAR A BASE DE DADOS	5
3	Views exclusivos ao utilizador userAPI	7
3.1	Criação do utilizador	7
3.2	Criação das Views	7
4	Stored Procedures	13
4.1	Procedimento para gerir dados da tabela central	13
4.2	Procedimentos com instruções dinâmicas	14
4.3	Procedimento que transforma um conjunto de dados de uma tabela num formato JSON	15
4.4	Procedimento que deverá validar e retornar uma informação	16
4.5	Procedimento que insira numa tabela baseado num dataset no for- mato JSON	17
5	Cursors	21
5.1	Cursor read only	21
5.2	Cursor de Update	22
6	Triggers	25
6.1	Trigger de validação	25
6.2	Trigger com notificação por email	26
6.3	Trigger que dispara após ser criado, alterado ou eliminado um registo.	32
6.4	Tabela de logs de todas as interações da base de dados alimentada com triggers	34
7	Instruções SQL	41
7.1	Criação de um Índice	41
7.2	Function	43
7.3	PIVOT	45
7.4	Consultas Simples	47
8	Otimização	49
8.1	Configuração do Tracer	49
8.2	Analisar desempenho da base de dados	53
8.3	Melhorar desempenho da base de dados	57
9	Relatório gerado pelo Report Builder	61
10	Rownumber, rank, denserank e partition	69
10.1	Aplicar o rownumber, rank, denserank e partition	69

11 Database Maintenance	71
11.1 Database Maintenance	71
12 CONCLUSÕES	77

Lista de Figuras

2.1	SQL Server Configuration Manager	3
2.2	SQL Server Configuration Manager	4
2.3	SQL Server Configuration Manager	4
2.4	Criar base de dados	5
2.5	Criar base de dados	5
2.6	Criar base de dados	6
2.7	Criar base de dados	6
3.1	Criação de um utilizador	7
3.2	Criação de uma view	7
3.3	Criação de uma view	8
3.4	Atribuição do privilegio ao userAPI	8
3.5	Atribuição do privilegio ao userAPI	8
3.6	Execução da view11	9
3.7	Execução da DataView	9
3.8	Execução da DataView	10
3.9	Execução da DataView	10
3.10	Execução da DataView	11
3.11	Execução da DataView	11
4.1	Código do procedimento para gerir dados da tabela central	13
4.2	Código do procedimento para gerir dados das tabelas de referência . .	14
4.3	Código do procedimento para gerir dados das tabelas de referência (2)	15
4.4	Código para tranformar/executar uma tabela com dados em ficheiro json	15
4.5	Resultado da transformação	15
4.6	Código do procedimento	16
4.7	Código do procedimento	17
4.8	Código do procedimento	18
4.9	Resultado do procedimento	19
5.1	Código cursor read only	21
5.2	Visualizar cursor read only	21
5.3	Código cursor read only	22
5.4	Visualizar cursor read only	22
5.5	Locais de residência repeditos	22
5.6	Número de registos com locais de residência repeditos	23
5.7	Código do procedimento para gerir dados das tabelas de referência . .	23
5.8	Verificar resultado depois da execução do cursor	23
6.1	Código do trigger de validação antes da inserção	25
6.2	Erro na inserção	25
6.3	Inserção válida	25
6.4	Welcome to Database Mail Configuration Wizard	26

6.5	Select Configuration Task	26
6.6	New Profile	27
6.7	New Database Mail Account	27
6.8	New Profile	28
6.9	Manage Profile Security	28
6.10	Configure System Parameters	29
6.11	Complete the Wizard	29
6.12	Configuração concluída	30
6.13	SQL Server Agent Properties	30
6.14	New Operator	31
6.15	Código do trigger com notificação por email	31
6.16	Email recebido	31
6.17	Código do trigger	33
6.18	Código da tabela logAtualizacoes	33
6.19	Código teste do trigger	33
6.20	Visualizar o movimento na tabela logAtualizacoes	33
6.21	Código do trigger	34
6.22	Código do trigger	34
6.23	Código do trigger	35
6.24	Código do trigger	35
6.25	Código do trigger	35
6.26	Código do trigger	36
6.27	Código do trigger	36
6.28	Código do trigger	36
6.29	Código do trigger	36
6.30	Código do trigger	37
6.31	Código do trigger	37
6.32	Código do trigger	37
6.33	Código do trigger	38
6.34	Código do trigger	38
6.35	Código do trigger	38
6.36	Código do trigger	39
6.37	Código do trigger	39
6.38	Código do trigger	39
6.39	Código do trigger	39
6.40	Código do trigger	40
6.41	Código do trigger	40
6.42	Código do trigger	40
7.1	Índice	41
7.2	Índice	41
7.3	Índice	42
7.4	Function	43
7.5	Function	44
7.6	Pivot	45
7.7	Pivot	46
7.8	Consulta SQL	47
7.9	Consulta SQL	47
7.10	Consulta SQL	48
7.11	Consulta SQL	48

8.1	Tracer Properties	49
8.2	Tracer Properties	49
8.3	Escolher Database	50
8.4	Verificar que o Tracer está a correr	50
8.5	Query	51
8.6	Verificar que a query passou no tracer	51
8.7	Parar o tracer	52
8.8	Exportar tracer	52
8.9	Database Engine Tuning Advisor	53
8.10	Aba Tuning Options	54
8.11	Advanced Tuning Options	54
8.12	Começar a análise	54
8.13	Recomendações	55
8.14	Reports	56
8.15	Guardar Recomendações	56
8.16	Analisar plano de execução da query	57
8.17	Verificar que há índices em falta	57
8.18	Observar Estimated Subtree Cost	58
8.19	Script para aplicar as recomendações	58
8.20	Observar que já não há falta de índices	59
9.1	Conectar á instancia	62
9.2	Status do servidor	62
9.3	Criar novo relatório	62
9.4	Criar um novo dataset	63
9.5	Conecção á base de dados	63
9.6	Selecionar dataset criado	64
9.7	Seleção das tabelas que se vão utilizar	64
9.8	Separar variáveis	65
9.9	Formato da tabela	65
9.10	Relatório gerado	66
9.11	Continuação Relatório gerado	66
10.1	Código do select	69
11.1	Maintenance Plan Wizard	71
11.2	Maintenance Plan Wizard	71
11.3	Maintenance Plan Wizard	72
11.4	Maintenance Plan Wizard	72
11.5	Maintenance Plan Wizard	73
11.6	Maintenance Plan Wizard	73
11.7	Maintenance Plan Wizard	74
11.8	Maintenance Plan Wizard	74
11.9	Maintenance Plan criado	75
11.10	Propriedade do job	75
11.11	Configurar job	76

Capítulo 1

INTRODUÇÃO

O projeto em questão visa desenvolver um sistema robusto de gestão de dados demográficos, utilizando um conjunto de dados provenientes dos Censos, que detalha a população residente com 15 anos ou mais por local de residência, sexo, grupo etário, condição perante o trabalho e nível de escolaridade mais elevado completo. Este trabalho foca na implementação e na gestão de uma base de dados que não só armazena, mas também manipula e analisa esses dados de maneira eficiente através de várias técnicas e ferramentas SQL.

Capítulo 2

FILESTREAM E PROCEDIMENTOS DE INSERÇÃO

2.1 HABILITAR FILESTREAM NO SSCM

Para implementar o filestream e procedimentos de inserção é preciso ativar o filestream no SQL Server configuration manager primeiro.

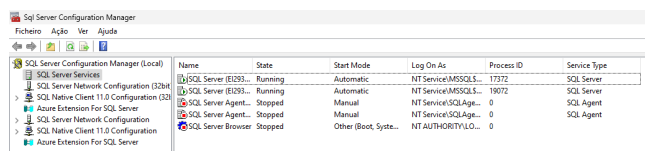


FIGURA 2.1: Propriedades do Servidor

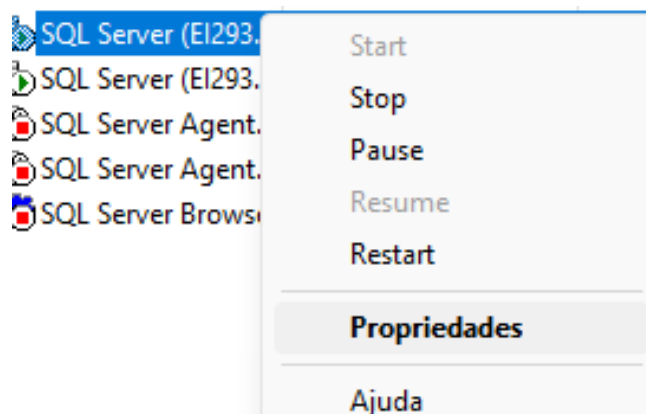


FIGURA 2.2: Propriedades do Servidor

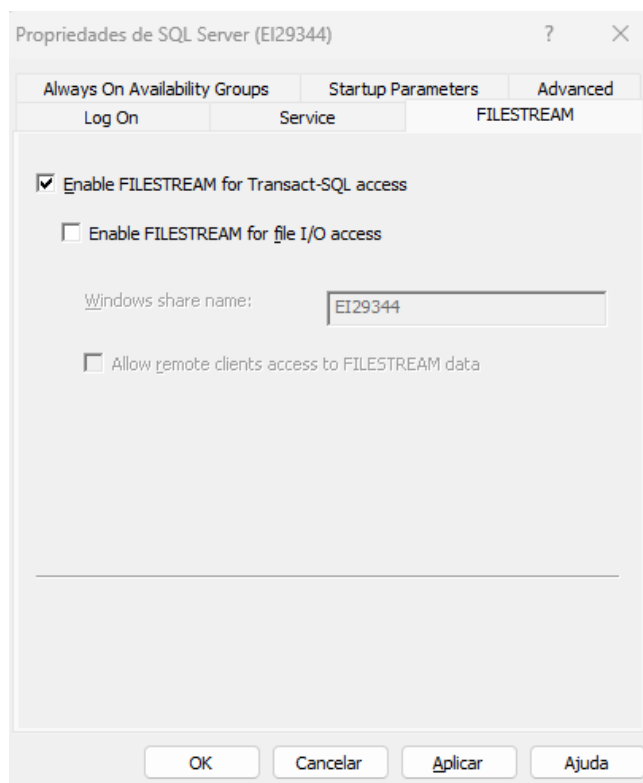


FIGURA 2.3: Filestream ativado

2.2 CRIAR A BASE DE DADOS

Após a ativação criamos uma base de dados chamado FileStreamDB no SQL Server com capacidades FileStream. O comando define o grupo de arquivos primário, nomeando o arquivo principal FileStreamDB e especificando seu caminho. Em seguida, criamos um novo grupo de arquivos chamado FileStreamGroup para conter dados FILESTREAM, especificando o arquivo FileStreamDB FileStream para armazenar esses dados. O arquivo de log do banco de dados é nomeado FileStreamDB Log e o seu caminho é definido.

```
SQLQuery1.sql - DE...eStreamDB (sa (63)) * > X
GO
--CREATE DATABASE FileStreamDB
ON
PRIMARY (NAME = FileStreamDB, FILENAME = 'C:\E-1\24Ano\25Semestre\ABD\Projeto\FileStreamDB.ndf'),
FILEGROUP FileStreamGroup CONTAINS FileStream (NAME = FileStreamDB_FileStream, FILENAME = 'C:\E-1\24Ano\25Semestre\ABD\Projeto\FileStreamDB_FileStream')
LOG ON (NAME = FileStreamDB_Log, FILENAME = 'C:\E-1\24Ano\25Semestre\ABD\Projeto\FileStreamDB.ldf');
GO

-- Messages
Comando completed successfully.
Completion time: 2024-06-01T14:46:29.5464762+01:00
```

FIGURA 2.4: Criar base de dados

Os comandos SQL criam várias tabelas na base de dados FileStreamDB. Primeiro, o comando USE FileStreamDB seleciona a base de dados. Em seguida, são criadas as tabelas Ano, localresidencia, sexo, grupoetario, condicaotrab e nivelescolaridade, cada uma com uma chave primária e uma coluna de descrição. Por fim, a tabela dados é criada com um identificador único id dados como chave primária, uma coluna valor, e várias chaves estrangeiras que referenciam as outras tabelas (id localre, id sexo, id grupoetario, id condicaotrab, id nivelescolaridade, id ano). A tabela dados também inclui uma coluna Conteudo do tipo VARBINARY(MAX) com capacidade FILESTREAM.

```
SQLQuery1.sql - DE...eStreamDB (sa (63)) * > X
USE FileStreamDB;
GO
--
CREATE TABLE Ano (
    id_ano INT PRIMARY KEY,
    ano INT
);
GO

CREATE TABLE localresidencia (
    id_localre INT PRIMARY KEY,
    descricao NVARCHAR(255)
);
GO

CREATE TABLE sexo (
    id_sexo INT PRIMARY KEY,
    descricao NVARCHAR(255)
);
GO

CREATE TABLE grupoetario (
    id_grupoetario INT PRIMARY KEY,
    descricao NVARCHAR(255)
);
GO

CREATE TABLE condicaotrab (
    id_condicaotrab INT PRIMARY KEY,
    descricao NVARCHAR(255)
);
GO

CREATE TABLE nivelescolaridade (
    id_nivelescolaridade INT PRIMARY KEY,
    descricao NVARCHAR(255)
);
GO

CREATE TABLE dados (
    id_dados UNIQUEIDENTIFIER ROWGUIDCOL NOT NULL PRIMARY KEY,
    valor NVARCHAR(255),
    id_localre INT FOREIGN KEY REFERENCES localresidencia(id_localre),
    id_sexo INT FOREIGN KEY REFERENCES sexo(id_sexo),
    id_grupoetario INT FOREIGN KEY REFERENCES grupoetario(id_grupoetario),
    id_condicaotrab INT FOREIGN KEY REFERENCES condicaotrab(id_condicaotrab),
    id_nivelescolaridade INT FOREIGN KEY REFERENCES nivelescolaridade(id_nivelescolaridade),
    id_ano INT FOREIGN KEY REFERENCES Ano(id_ano),
    Conteudo VARBINARY(MAX) FILESTREAM NULL
);
```

FIGURA 2.5: Criar as tabelas

O procedimento cria uma stored procedure chamada InserirDados na base de dados FileStreamDB. A stored procedure aceita parâmetros de entrada (@valor, @id localre, @id sexo, @id grupoeatrio, @id condicaotrab, @id nivelescolaridade, @id ano, @Conteudo) e insere um novo registro na tabela dados usando esses valores. A variável @id dados é declarada como um identificador único e inicializada com um novo valor NEWID. A stored procedure insere os dados na tabela dados e retorna o valor do identificador único gerado (@id dados) como DocumentoID.

```
SQLQuery1.sql - FileStreamDB (sa [SQL])
-- CREATE PROCEDURE InserirDados
-- (
--     @valor NVARCHAR(255),
--     @id_localre INT,
--     @id_sexo INT,
--     @id_grupoetario INT,
--     @id_condicaotrab INT,
--     @id_nivelescolaridade INT,
--     @id_ano INT,
--     @Conteudo VARBINARY(MAX)
-- )
-- AS
-- BEGIN
--     SET NOCOUNT ON;
--
--     DECLARE @id_dados UNIQUEIDENTIFIER = NEWID();
--
--     INSERT INTO dados (id_dados, valor, id_localre, id_sexo, id_grupoetario, id_condicaotrab, id_nivelescolaridade, id_ano, Conteudo)
--     VALUES (@id_dados, @valor, @id_localre, @id_sexo, @id_grupoetario, @id_condicaotrab, @id_nivelescolaridade, @id_ano, @Conteudo);
--
--     SELECT @id_dados AS DocumentoID;
-- END
-- GO
```

FIGURA 2.6: Criar o procedimentos

O código SQL declara uma variável @DocumentContent do tipo VARBINARY(MAX) e atribui a ela o valor 'Teste de conteúdo de arquivo' convertido para VARBINARY(MAX). Em seguida, executa a stored procedure InserirDados com parâmetros específicos, incluindo o valor 'Valor de Teste' e outros identificadores, bem como o conteúdo da variável @DocumentContent. Após a execução da stored procedure, realiza uma consulta na tabela dados para recuperar e exibir os dados inseridos, convertendo o campo Conteudo de VARBINARY(MAX) para NVARCHAR(MAX). O resultado da execução inclui o identificador do documento inserido.

```
SQLQuery1.sql - FileStreamDB (sa [SQL])
-- DECLARE @DocumentContent VARBINARY(MAX);
-- -- Simulação de um conteúdo de arquivo em VARBINARY
-- SET @DocumentContent = CAST('Teste de conteúdo de arquivo' AS VARBINARY(MAX));
-- EXEC InserirDados
--     @valor = 'Valor de Teste',
--     @id_localre = 1,
--     @id_sexo = 1,
--     @id_grupoetario = 1,
--     @id_condicaotrab = 1,
--     @id_nivelescolaridade = 1,
--     @id_ano = 1,
--     @Conteudo = @DocumentContent;
-- GO
-- -- Consultar os dados inseridos
-- SELECT id_dados, valor, id_localre, id_sexo, id_grupoetario, id_condicaotrab, id_nivelescolaridade, id_ano, CAST(Conteudo AS NVARCHAR(MAX)) AS Conteudo
-- FROM dados;
-- GO
```

DocumentoID
1

FIGURA 2.7: Consulta o identificador do documento inserido

Capítulo 3

Views exclusivos ao utilizador userAPI

3.1 Criação do utilizador

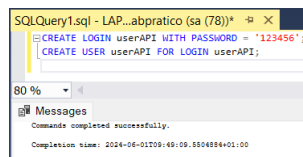


FIGURA 3.1: Código para criar utilizador

3.2 Criação das Views

```
CREATE VIEW view11 AS
SELECT
    a.ano,
    lr.descricao AS local_residencia,
    ne.descricao AS nivel_escolaridade,
    COUNT(d.id_dado) AS total_registros
FROM
    dados d
    INNER JOIN ano a ON d.id_ano = a.id_ano
    INNER JOIN localresidencia lr ON d.id_localre = lr.id_localre
    INNER JOIN nivelescolaridade ne ON d.id_nivelescolaridade = ne.id_nivelescolaridade
GROUP BY
    a.ano,
    lr.descricao,
    ne.descricao
HAVING
    a.ano = 2011;
```

FIGURA 3.2: Código de uma view

Com esta view vai ser possível visualizar a quantidade de registos por local de residência e nível de escolaridade para o ano de 2011.

```
CREATE VIEW viewvalor AS
SELECT
  lr.descricao AS LocalResidencia,
  ge.descricao AS GrupoEtario,
  ne.descricao AS NivelEscolaridade,
  COUNT(d.id_dado) AS TotalRegistros,
  SUM(d.valor) AS SomaValor,
  AVG(d.valor) AS MediaValor,
  MAX(d.valor) AS ValorMaximo,
  MIN(d.valor) AS ValorMinimo
FROM
  dados d
  INNER JOIN localresidencia lr ON d.id_localre = lr.id_localre
  INNER JOIN grupoetario ge ON d.id_grupoetario = ge.id_grupoetario
  INNER JOIN nivelescolaridade ne ON d.id_nivelescolaridade = ne.id_nivelescolaridade
GROUP BY
  lr.descricao,
  ge.descricao,
  ne.descricao;
```

FIGURA 3.3: Código de uma view

Com esta view é possível visualizar o total de registos, soma/media/valorMax/valorMin do parametro valor para os registos dos diferentes tipos de residencia, grupo etário e nivel escolaridade

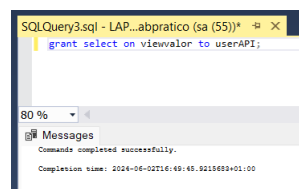


FIGURA 3.4: Código para atribuir o privilegio ao userAPI

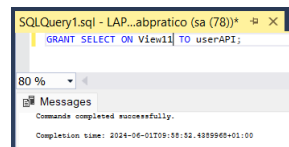


FIGURA 3.5: Código para atribuir o privilegio ao userAPI

SQLQuery1.sql - LAP...abpratico (sa (78)) * X

```
SELECT * FROM view11;
```

80 %

Results Messages

	ano	local_residencia	nivel_escolaridade	total_registr
1	2011	Continente	Ensino pos-secundario	840
2	2011	Norte	Ensino secundario	840
3	2011	Algarve	Ensino basico	420
4	2011	Centro	Ensino basico	420
5	2011	Centro	Mestrado	420
6	2011	Area Metropolitana de Lisboa	Ensino basico	840
7	2011	Area Metropolitana de Lisboa	Curso tecnico superior profissional	840
8	2011	Area Metropolitana de Lisboa	Mestrado	840
9	2011	Douro	3 Ciclo	420
10	2011	Douro	Ensino superior	420
11	2011	Douro	Doutoramento	420
12	2011	Alentejo Central	Total	420
13	2011	Regiao Autonoma dos Acores	1 Ciclo	1260
14	2011	Area Metropolitana do Porto	Bacharelato	828
15	2011	Regiao de Leiria	Total	420
16	2011	Viseu Dao Lafoes	1 Ciclo	420
17	2011	Oeste	Ensino superior	420
18	2011	Oeste	3 Ciclo	420
19	2011	Alentejo Litoral	Doutoramento	420
20	2011	Alentejo Litoral	Curso tecnico superior profissional	420
21	2011	Portugal	1 Ciclo	420
22	2011	Portugal	Bacharelato	420
23	2011	Cavado	Total	840
24	2011	Ave	Bacharelato	840
25	2011	Ave	Total	840
26	2011	Regiao de Coimbra	Mestrado	420
27	2011	Regiao de Coimbra	Ensino basico	420
28	2011	Regiao de Leiria	Licenciatura	420
29	2011	Viseu Dao Lafoes	Ensino basico	420
30	2011	Viseu Dao Lafoes	3 Ciclo	420
31	2011	Viseu Dao Lafoes	Ensino superior	420
32	2011	Regiao Autonoma dos Acores	Ensino basico	1260

FIGURA 3.6: Visualização da view criada

Results Messages

	LocalResidencia	GrupoEtario	NivelEscolaridade	TotalRegistros	SomaValor	MediaValor	ValorMaximo	ValorMinimo
1	Alto Minho	15 - 19 anos	Licenciatura	100	0	0	0	0
2	Area Metropolitana de Lisboa	65 - 69 anos	Ensino pos-secundario	120	1236	10.3	103	0
3	Douro	75 ou mais anos	Bacharelato	60	3876	64.6	401	0
4	Alto Minho	30 - 34 anos	Ensino superior	100	68304	683.04	3638	0
5	Alto Alentejo	40 - 44 anos	Mestrado	60	1914	31.9	229	0
6	Viseu Dao Lafoes	50 - 54 anos	Licenciatura	60	21396	356.6	2207	0
7	Douro	65 - 69 anos	Ensino secundario	60	7296	121.6	964	0
8	Alentejo	75 ou mais anos	Ensino secundario	60	19536	325.6	2150	0
9	Regiao Autonoma da Madeira	65 - 69 anos	Ensino basico	180	336330	1868.5	11061	0
10	Area Metropolitana do Porto	45 - 49 anos	2 Ciclo	90	504288	5603.2	29068	5
11	Area Metropolitana do Porto	65 - 69 anos	Doutoramento	90	5532	61.4666666666667	570	0
12	Centro	35 - 39 anos	Ensino superior	60	478920	7982	42586	4
13	Algarve	75 ou mais anos	Licenciatura	60	11754	195.9	1390	0
14	Area Metropolitana de Lisboa	75 ou mais anos	Ensino pos-secundario	120	1236	10.3	103	0
15	Baixo Alentejo	65 - 69 anos	Ensino secundario	60	5358	89.3	719	0
16	Viseu Dao Lafoes	Total	Bacharelato	60	35862	597.7	3462	0
17	Beira Baixa	50 - 54 anos	Ensino superior	60	10632	177.2	1060	0
18	Alto Alentejo	20 - 24 anos	Licenciatura	60	9462	157.7	901	0
19	Ave	65 - 69 anos	Curso tecnico superior profissional	90	0	0	0	0
20	Beira Baixa	70 - 74 anos	3 Ciclo	60	5358	89.3	701	0
21	Continente	35 - 39 anos	1 Ciclo	90	869142	9657.133333333333	66557	14
22	Alentejo Litoral	30 - 34 anos	Doutoramento	60	132	2.2	13	0
23	Regiao de Aveiro	35 - 39 anos	Ensino pos-secundario	60	6936	115.6	630	0
24	Beiras e Serras da Estrela	45 - 49 anos	Doutoramento	60	1260	21	138	0
25	Douro	40 - 44 anos	1 Ciclo	60	24180	403	3200	0
26	Baixo Alentejo	70 - 74 anos	1 Ciclo	60	48972	816.2	4412	0
27	Alto Minho	65 - 69 anos	3 Ciclo	100	18918	189.18	1491	0
28	Alentejo	45 - 49 anos	1 Ciclo	60	96378	1606.3	12179	0
29	Ave	15 - 19 anos	Ensino basico	90	345834	3842.6	22207	0
30	Regiao de Coimbra	50 - 54 anos	Nenhum	60	14028	233.8	1501	0

FIGURA 3.7: Visualização da view criada

A VIEW `dbo.vw.dados.simples` é usada para fornecer uma visão simplificada dos dados combinados de várias tabelas. Ela traz todas as descrições relevantes (localidade, sexo, grupo etário, condição de trabalho, nível de escolaridade, ano) junto com o identificador dos dados (`id.dados`) e o valor associado.

```

Instruções.sql - DE...ABD_censos (sa (83)) - X
-- Instrução SQL - View
CREATE VIEW dbo.vw_dados_simples AS
SELECT
    d.id_dados,
    lr.descricao AS localidade,
    s.descricao AS sexo,
    ge.descricao AS grupo_etario,
    ct.descricao AS condicao_trabalho,
    ne.descricao AS nivel_escolaridade,
    a.ano,
    d.valor
FROM
    dbo.dados d
JOIN
    dbo.localresidencia lr ON d.id_localre = lr.id_localre
JOIN
    dbo.sexo s ON d.id_sexo = s.id_sexo
JOIN
    dbo.grupoetario ge ON d.id_grupoetario = ge.id_grupoetario
JOIN
    dbo.condicaotrab ct ON d.id_condicaotrab = ct.id_condicaotrab
JOIN
    dbo.nivelescolaridade ne ON d.id_nivelescolaridade = ne.id_nivelescolaridade
JOIN
    dbo.ano a ON d.id_ano = a.id_ano;
SELECT *
FROM vw_dados_simples;

```

FIGURA 3.8: Código de uma View

id_dados	localidade	sexo	grupo_etario	condicao_trabalho	nivel_escolaridade	ano	valor
423259	Portugal	HM	Total	Desempregados	Licenciatura	2021	48919
423260	Portugal	HM	Total	Desempregados	Mestrado	2021	14886
423261	Portugal	HM	Total	Desempregados	Doutoramento	2021	997
423262	Portugal	HM	Total	Desempregados	Total	2021	391...
423263	Portugal	HM	Total	População inativa	Nenhum	2021	471...
423264	Portugal	HM	Total	População inativa	Ensino básico	2021	264...
423265	Portugal	HM	Total	População inativa	1.º ciclo	2021	158...
423266	Portugal	HM	Total	População inativa	2.º ciclo	2021	343...
423267	Portugal	HM	Total	População inativa	3.º ciclo	2021	720...
423268	Portugal	HM	Total	População inativa	Ensino secundário	2021	704...
423269	Portugal	HM	Total	População inativa	Ensino pós-secu...	2021	15
423270	Portugal	HM	Total	População inativa	Ensino superior	2021	370...
423271	Portugal	HM	Total	População inativa	Curso técnico su...	2021	4494
423272	Portugal	HM	Total	População inativa	Bacharelato	2021	63569
423273	Portugal	HM	Total	População inativa	Licenciatura	2021	247...
423274	Portugal	HM	Total	População inativa	Mestrado	2021	45374
423275	Portugal	HM	Total	População inativa	Doutoramento	2021	9645
423276	Portugal	HM	Total	População inativa	Total	2021	419...
423277	Portugal	HM	Total	Estudantes	Nenhum	2021	2716
423278	Portugal	HM	Total	Estudantes	Ensino básico	2021	311...
423279	Portugal	HM	Total	Estudantes	1.º ciclo	2021	1531
423280	Portugal	HM	Total	Estudantes	2.º ciclo	2021	36321
423281	Portugal	HM	Total	Estudantes	3.º ciclo	2021	273...
423282	Portugal	HM	Total	Estudantes	Ensino secundário	2021	281...
423283	Portugal	HM	Total	Estudantes	Ensino pós-secu...	2021	1
423284	Portugal	HM	Total	Estudantes	Ensino superior	2021	81191
423285	Portugal	HM	Total	Estudantes	Curso técnico su...	2021	2574
423286	Portugal	HM	Total	Estudantes	Bacharelato	2021	131
423287	Portugal	HM	Total	Estudantes	Licenciatura	2021	65265
423288	Portugal	HM	Total	Estudantes	Mestrado	2021	12610
423289	Portugal	HM	Total	Estudantes	Doutoramento	2021	611
423290	Portugal	HM	Total	Estudantes	Total	2021	676...
423291	Portugal	HM	Total	Domésticos	Nenhum	2021	24607
423292	Portugal	HM	Total	Domésticos	Ensino básico	2021	209...
423293	Portugal	HM	Total	Domésticos	1.º ciclo	2021	116...
423294	Portugal	HM	Total	Domésticos	2.º ciclo	2021	45622
423295	Portugal	HM	Total	Domésticos	3.º ciclo	2021	46814
423296	Portugal	HM	Total	Domésticos	Ensino secundário	2021	48494
423297	Portugal	HM	Total	Domésticos	Ensino pós-secu...	2021	0
423298	Portugal	HM	Total	Domésticos	Ensino superior	2021	22138
423299	Portugal	HM	Total	Domésticos	Curso técnico su...	2021	156

FIGURA 3.9: Visualização da view criada

A VIEW `dbo.vw.distribuicao.dados` é utilizada para agregar dados e fornecer estatísticas resumidas, como o número total de registos, a média dos valores e a soma dos valores, para diferentes combinações de localidade, sexo, grupo etário, condição de trabalho, nível de escolaridade e ano. Faz isto combinando várias tabelas de descrição e valores, agrupando os dados e filtrando grupos que têm mais de um registo.

```

Instruções.sql - DE...ABD_censos (sa (83))
-- Instrução SQL - View
CREATE VIEW dbo.vw_distribuicao_dados_AS
SELECT
    lr.descricao AS localidade,
    s.descricao AS sexo,
    ge.descricao AS grupo_etario,
    ct.descricao AS condicao_trabalho,
    ne.descricao AS nivel_escolaridade,
    a.ano,
    COUNT(d.id_dados) AS total_registros,
    AVG(d.valor) AS media_valores,
    SUM(d.valor) AS soma_valores
FROM
    dbo.dados d
JOIN
    dbo.localresidencia lr ON d.id_localre = lr.id_localre
JOIN
    dbo.sexo s ON d.id_sexo = s.id_sexo
JOIN
    dbo.grupoetario ge ON d.id_grupoetario = ge.id_grupoetario
JOIN
    dbo.condicaotrab ct ON d.id_condicaotrab = ct.id_condicaotrab
JOIN
    dbo.nivelescolaridade ne ON d.id_nivelescolaridade = ne.id_nivelescolaridade
JOIN
    dbo.ano a ON d.id_ano = a.id_ano
GROUP BY
    lr.descricao, s.descricao, ge.descricao, ct.descricao, ne.descricao, a.ano
HAVING
    COUNT(d.id_dados) > 1;
SELECT * FROM vw_distribuicao_dados;

```

FIGURA 3.10: Código de uma View

localidade	sexo	grupo_etario	condicao_trabalho	nivel_escolaridade	ano	total_registros	media_valores	soma_valores
Algarve	HM	15 - 19 anos	Desempregados	1.º ciclo	2012	2	43	86
Algarve	HM	15 - 19 anos	Desempregados	2.º ciclo	2021	2	46	92
Algarve	HM	15 - 19 anos	Desempregados	3.º ciclo	2012	2	857	1714
Algarve	HM	15 - 19 anos	Desempregados	1.º ciclo	2021	2	6	12
Algarve	HM	15 - 19 anos	Desempregados	Curso técnico superior profissional	2012	2	0	0
Algarve	HM	15 - 19 anos	Desempregados	Curso técnico superior profissional	2021	2	2	4
Algarve	HM	15 - 19 anos	Desempregados	3.º ciclo	2021	2	220	440
Algarve	HM	15 - 19 anos	Desempregados	Ensino básico	2012	2	1191	2382
Algarve	HM	15 - 19 anos	Desempregados	Ensino básico	2021	2	272	544
Algarve	HM	15 - 19 anos	Desempregados	Ensino pós-secundário	2012	2	17	34
Algarve	HM	15 - 19 anos	Desempregados	Bacharelato	2012	2	0	0
Algarve	HM	15 - 19 anos	Desempregados	Bacharelato	2021	2	0	0
Algarve	HM	15 - 19 anos	Desempregados	Ensino secundário	2012	2	292	584
Algarve	HM	15 - 19 anos	Desempregados	2.º ciclo	2012	2	291	582
Algarve	HM	15 - 19 anos	Desempregados	Doutoramento	2012	2	0	0
Algarve	HM	15 - 19 anos	Desempregados	Licenciatura	2021	2	0	0
Algarve	HM	15 - 19 anos	Desempregados	Doutoramento	2021	2	0	0
Algarve	HM	15 - 19 anos	Desempregados	Ensino secundário	2021	2	241	482
Algarve	HM	15 - 19 anos	Desempregados	Mestrado	2012	2	0	0
Algarve	HM	15 - 19 anos	Desempregados	Ensino pós-secundário	2021	2	0	0
Algarve	HM	15 - 19 anos	Desempregados	Nenhum	2012	2	25	50
Algarve	HM	15 - 19 anos	Desempregados	Licenciatura	2012	2	0	0
Algarve	HM	15 - 19 anos	Desempregados	Nenhum	2021	2	11	22
Algarve	HM	15 - 19 anos	Desempregados	Mestrado	2021	2	0	0
Algarve	HM	15 - 19 anos	Domésticos	2.º ciclo	2021	2	8	16
Algarve	HM	15 - 19 anos	Desempregados	Ensino superior	2012	2	0	0
Algarve	HM	15 - 19 anos	Domésticos	3.º ciclo	2012	2	74	148
Algarve	HM	15 - 19 anos	Domésticos	Bacharelato	2012	2	0	0
Algarve	HM	15 - 19 anos	Domésticos	2.º ciclo	2012	2	47	94
Algarve	HM	15 - 19 anos	Desempregados	Total	2012	2	1525	3050
Algarve	HM	15 - 19 anos	Domésticos	Curso técnico superior profissional	2012	2	0	0
Algarve	HM	15 - 19 anos	Domésticos	Curso técnico superior profissional	2021	2	0	0
Algarve	HM	15 - 19 anos	Desempregados	Total	2021	2	526	1052
Algarve	HM	15 - 19 anos	Domésticos	1.º ciclo	2021	2	7	14
Algarve	HM	15 - 19 anos	Domésticos	Ensino básico	2021	2	53	106
Algarve	HM	15 - 19 anos	Domésticos	Doutoramento	2012	2	0	0
Algarve	HM	15 - 19 anos	Domésticos	Ensino pós-secundário	2021	2	0	0
Algarve	HM	15 - 19 anos	Domésticos	Bacharelato	2021	2	0	0
Algarve	HM	15 - 19 anos	Domésticos	Ensino secundário	2012	2	26	52
Algarve	HM	15 - 19 anos	Domésticos	Ensino superior	2012	2	0	0
Algarve	HM	15 - 19 anos	Desempregados	Ensino superior	2021	2	2	4
Algarve	HM	15 - 19 anos	Domésticos	Ensino básico	2012	2	154	308

FIGURA 3.11: Visualização da view criada

Esta VIEW facilita a execução de consultas sobre os dados agregados, permitindo que os utilizadores obtenham estatísticas resumidas sem terem de escrever a consulta completa repetidamente.

Capítulo 4

Stored Procedures

4.1 Procedimento para gerir dados da tabela central

```

CREATE PROCEDURE gerirDados
    @id_dados int,
    @id_localre VARCHAR(50) = NULL,
    @id_sexo VARCHAR(50) = NULL,
    @id_grupoetario VARCHAR(50) = NULL,
    @id_nivelescolaridade VARCHAR(50) = NULL,
    @id_condicaotrab VARCHAR(50) = NULL,
    @id_ano VARCHAR(50) = NULL,
    @valor FLOAT = NULL,
    @operacao INT
AS
BEGIN
    BEGIN TRANSACTION;

    IF @operacao IN (1, 2) AND NOT EXISTS (SELECT 1 FROM dbo.dados WHERE id_dados = @id_dados)
    BEGIN
        ROLLBACK TRANSACTION;
        RETURN;
    END

    IF @operacao = 0
    BEGIN
        IF @id_localre IS NOT NULL AND @id_sexo IS NOT NULL AND @id_grupoetario IS NOT NULL
            AND @id_nivelescolaridade IS NOT NULL AND @id_condicaotrab IS NOT NULL
            AND @id_ano IS NOT NULL AND @valor IS NOT NULL
        BEGIN
            INSERT INTO dbo.dados(valor, id_localre, id_sexo, id_grupoetario, id_condicaotrab, id_nivelescolaridade, id_ano)
            VALUES (@valor, @id_localre, @id_sexo, @id_grupoetario, @id_condicaotrab, @id_nivelescolaridade, @id_ano);
        END
        ELSE
        BEGIN
            ROLLBACK TRANSACTION;
            RETURN;
        END
    END

    ELSE IF @operacao = 1
    BEGIN
        UPDATE dbo.dados
        SET
            id_localre = CASE WHEN @id_localre IS NOT NULL THEN @id_localre ELSE id_localre END,
            id_sexo = CASE WHEN @id_sexo IS NOT NULL THEN @id_sexo ELSE id_sexo END,
            id_grupoetario = CASE WHEN @id_grupoetario IS NOT NULL THEN @id_grupoetario ELSE id_grupoetario END,
            id_nivelescolaridade = CASE WHEN @id_nivelescolaridade IS NOT NULL THEN @id_nivelescolaridade ELSE id_nivelescolaridade END,
            id_condicaotrab = CASE WHEN @id_condicaotrab IS NOT NULL THEN @id_condicaotrab ELSE id_condicaotrab END,
            id_ano = CASE WHEN @id_ano IS NOT NULL THEN @id_ano ELSE id_ano END,
            valor = CASE WHEN @valor IS NOT NULL THEN @valor ELSE valor END
        WHERE id_dados = @id_dados;
    END

    ELSE IF @operacao = 2
    BEGIN
        DELETE FROM dbo.dados WHERE id_dados = @id_dados;
    END

    COMMIT TRANSACTION;
END

```

FIGURA 4.1: Código do procedimento para gerir dados da tabela central

O procedimento acima têm como objetivo gerir os dados da tabela central, sendo assim os parametros são os campos da tabela central e o indicativo da operação a executar (0 - Insert, 1 - Update, 2 - Delete). Este procedimento controla uma transação e casos de erro, para isso no caso da operação ser update ou delete uma vez que é necessário o id ser válido a transação dá rollback no caso de não existir nenhum registo com esse id. Os parametros referentes aos campos da tabela dados são inicializados a nulo para que em caso de update seja apenas necessário preencher os parametros que vão ser alterados, uma vez que esta prática foi implementada em caso de insert é necessário validar que todos os parametros não são nulos exceto o id que é incrementado de forma automática snedo que o atributo é serial.

4.2 Procedimentos com instruções dinâmicas

Procedimentos com instruções dinâmicas são procedimentos que recebem por parametro algo que influencia diretamente a query do procedimento por exemplo, um select a uma tabela em que o nome da tabela é passado por parametro.

```

CREATE PROCEDURE gerirTabelasSecundarias
    @tabela VARCHAR(50),
    @id VARCHAR(50),
    @valor VARCHAR(50),
    @operacao INT
AS
BEGIN
    IF @operacao = 0
    BEGIN
        IF @tabela = 'ano'
        BEGIN
            INSERT INTO dbo.ano (id_ano, ano) VALUES (@id, @valor);
        END
        ELSE IF @tabela = 'condicaootrab'
        BEGIN
            INSERT INTO dbo.condicaootrab (id_condicaootrab, descricao) VALUES (@id, @valor);
        END
        ELSE IF @tabela = 'grupoetario'
        BEGIN
            INSERT INTO dbo.grupoetario (id_grupoetario, descricao) VALUES (@id, @valor);
        END
        ELSE IF @tabela = 'localresidencia'
        BEGIN
            INSERT INTO dbo.localresidencia (id_localre, descricao) VALUES (@id, @valor);
        END
        ELSE IF @tabela = 'nivelescolaridade'
        BEGIN
            INSERT INTO dbo.nivelescolaridade (id_nivelescolaridade, descricao) VALUES (@id, @valor);
        END
        ELSE IF @tabela = 'sexo'
        BEGIN
            INSERT INTO dbo.sexo (id_sexo, descricao) VALUES (@id, @valor);
        END
    END;

    IF @operacao = 1
    BEGIN
        IF @tabela = 'ano'
        BEGIN
            UPDATE dbo.ano SET ano = @valor, id_ano = 'STA' + @valor WHERE id_ano = @id;
        END
        ELSE IF @tabela = 'condicaootrab'
        BEGIN
            UPDATE dbo.condicaootrab SET descricao = @valor WHERE id_condicaootrab = @id;
        END
        ELSE IF @tabela = 'grupoetario'
        BEGIN
            UPDATE dbo.grupoetario SET descricao = @valor WHERE id_grupoetario = @id;
        END
        ELSE IF @tabela = 'localresidencia'
        BEGIN
            UPDATE dbo.localresidencia SET descricao = @valor WHERE id_localre = @id;
        END
        ELSE IF @tabela = 'nivelescolaridade'
        BEGIN
            UPDATE dbo.nivelescolaridade SET descricao = @valor WHERE id_nivelescolaridade = @id;
        END
        ELSE IF @tabela = 'sexo'
        BEGIN
            UPDATE dbo.sexo SET descricao = @valor WHERE id_sexo = @id;
        END
    END;
END;

```

FIGURA 4.2: Código do procedimento para gerir dados das tabelas de referência

Todas as tabelas de referência têm apenas dois campos, id e um valor. O procedimento recebe por parametros o nome da tabela, o id, o valor e a operação. Mediante o nome da tabela cada operação é tratada da devida forma.

```

IF @operacao = 2
BEGIN
    IF @tabela = 'ano'
    BEGIN
        DELETE FROM dbo.ano WHERE id_ano = @id;
    END
    ELSE IF @tabela = 'condicaootrab'
    BEGIN
        DELETE FROM dbo.condicaootrab WHERE id_condicaootrab = @id;
    END
    ELSE IF @tabela = 'grupoetario'
    BEGIN
        DELETE FROM dbo.grupoetario WHERE id_grupoetario = @id;
    END
    ELSE IF @tabela = 'localresidencia'
    BEGIN
        DELETE FROM dbo.localresidencia WHERE id_localre = @id;
    END
    ELSE IF @tabela = 'nivelescolaridade'
    BEGIN
        DELETE FROM dbo.nivelescolaridade WHERE id_nivelescolaridade = @id;
    END
    ELSE IF @tabela = 'sexo'
    BEGIN
        DELETE FROM dbo.sexo WHERE id_sexo = @id;
    END
END;
END;

```

FIGURA 4.3: Código do procedimento para gerir dados das tabelas de referência (2)

4.3 Procedimento que transforma um conjunto de dados de uma tabela num formato JSON

```

create procedure dataj as select top 1000 * from dbo.dados for json path
exec dataj;

```

FIGURA 4.4: Código para transformar os dados num ficheiro json

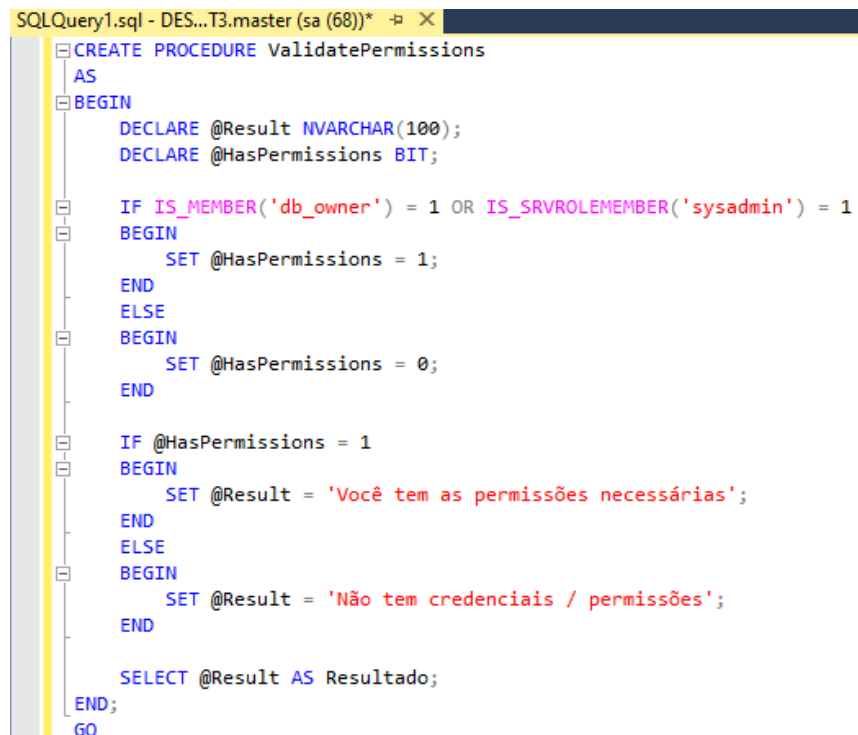
```

[{"id_dado":1,"id_ano":"57A2011","id_localre":"1","id_sexo":"1","id_grupoetario":"01","id_condicaootrab":"1","id_nivelescolaridade":"1","valor":3.740000000000000e+
002}, {"id_dado":2,"id_ano":"57A2011","id_localre":"1","id_sexo":"1","id_grupoetario":"01","id_condicaootrab":"1","id_nivelescolaridade":"2","valor":2.948700000000000e+
004}, {"id_dado":3,"id_ano":"57A2011","id_localre":"1","id_sexo":"1","id_grupoetario":"01","id_condicaootrab":"1","id_nivelescolaridade":"21","valor":1.219000000000000e+
003}, {"id_dado":4,"id_ano":"57A2011","id_localre":"1","id_sexo":"1","id_grupoetario":"01","id_condicaootrab":"1","id_nivelescolaridade":"22","valor":6.656000000000000e+
003}, {"id_dado":5,"id_ano":"57A2011","id_localre":"1","id_sexo":"1","id_grupoetario":"01","id_condicaootrab":"1","id_nivelescolaridade":"23","valor":2.161200000000000e+
004}, {"id_dado":6,"id_ano":"57A2011","id_localre":"1","id_sexo":"1","id_grupoetario":"01","id_condicaootrab":"1","id_nivelescolaridade":"3","valor":8.648000000000000e+
003}, {"id_dado":7,"id_ano":"57A2011","id_localre":"1","id_sexo":"1","id_grupoetario":"01","id_condicaootrab":"1","id_nivelescolaridade":"4","valor":5.190000000000000e+
002}, {"id_dado":8,"id_ano":"57A2011","id_localre":"1","id_sexo":"1","id_grupoetario":"01","id_condicaootrab":"1","id_nivelescolaridade":"5","valor":1.000000000000000e+
000}, {"id_dado":9,"id_ano":"57A2011","id_localre":"1","id_sexo":"1","id_grupoetario":"01","id_condicaootrab":"1","id_nivelescolaridade":"51","valor":0.000000000000000e+
000}, {"id_dado":10,"id_ano":"57A2011","id_localre":"1","id_sexo":"1","id_grupoetario":"01","id_condicaootrab":"1","id_nivelescolaridade":"52","valor":1.000000000000000e+
000}, {"id_dado":11,"id_ano":"57A2011","id_localre":"1","id_sexo":"1","id_grupoetario":"01","id_condicaootrab":"1","id_nivelescolaridade":"53","valor":0.000000000000000e+
000}, {"id_dado":12,"id_ano":"57A2011","id_localre":"1","id_sexo":"1","id_grupoetario":"01","id_condicaootrab":"1","id_nivelescolaridade":"54","valor":0.000000000000000e+
000}, {"id_dado":13,"id_ano":"57A2011","id_localre":"1","id_sexo":"1","id_grupoetario":"01","id_condicaootrab":"1","id_nivelescolaridade":"55","valor":1.642800000000000e+
000}, {"id_dado":14,"id_ano":"57A2011","id_localre":"1","id_sexo":"1","id_grupoetario":"01","id_condicaootrab":"1","id_nivelescolaridade":"17","valor":3.902900000000000e+
004}, {"id_dado":15,"id_ano":"57A2011","id_localre":"1","id_sexo":"1","id_grupoetario":"01","id_condicaootrab":"1","id_nivelescolaridade":"1","valor":1.800000000000000e+
002}, {"id_dado":16,"id_ano":"57A2011","id_localre":"1","id_sexo":"1","id_grupoetario":"01","id_condicaootrab":"1","id_nivelescolaridade":"2","valor":1.642800000000000e+
004}, {"id_dado":17,"id_ano":"57A2011","id_localre":"1","id_sexo":"1","id_grupoetario":"01","id_condicaootrab":"1","id_nivelescolaridade":"21","valor":5.270000000000000e+
002}, {"id_dado":18,"id_ano":"57A2011","id_localre":"1","id_sexo":"1","id_grupoetario":"01","id_condicaootrab":"1","id_nivelescolaridade":"22","valor":3.362000000000000e+
003}, {"id_dado":19,"id_ano":"57A2011","id_localre":"1","id_sexo":"1","id_grupoetario":"01","id_condicaootrab":"1","id_nivelescolaridade":"23","valor":1.253900000000000e+
004}, {"id_dado":20,"id_ano":"57A2011","id_localre":"1","id_sexo":"1","id_grupoetario":"01","id_condicaootrab":"1","id_nivelescolaridade":"5","valor":5.500000000000000e+
003}, {"id_dado":21,"id_ano":"57A2011","id_localre":"1","id_sexo":"1","id_grupoetario":"01","id_condicaootrab":"1","id_nivelescolaridade":"4","valor":3.230000000000000e+
002}, {"id_dado":22,"id_ano":"57A2011","id_localre":"1","id_sexo":"1","id_grupoetario":"01","id_condicaootrab":"1","id_nivelescolaridade":"5","valor":1.000000000000000e+
000}, {"id_dado":23,"id_ano":"57A2011","id_localre":"1","id_sexo":"1","id_grupoetario":"01","id_condicaootrab":"1","id_nivelescolaridade":"51","valor":0.000000000000000e+
000}, {"id_dado":24,"id_ano":"57A2011","id_localre":"1","id_sexo":"1","id_grupoetario":"01","id_condicaootrab":"1","id_nivelescolaridade":"52","valor":1.000000000000000e+
000}, {"id_dado":25,"id_ano":"57A2011","id_localre":"1","id_sexo":"1","id_grupoetario":"01","id_condicaootrab":"1","id_nivelescolaridade":"53","valor":0.000000000000000e+
000}, {"id_dado":26,"id_ano":"57A2011","id_localre":"1","id_sexo":"1","id_grupoetario":"01","id_condicaootrab":"1","id_nivelescolaridade":"54","valor":0.000000000000000e+
000}, {"id_dado":27,"id_ano":"57A2011","id_localre":"1","id_sexo":"1","id_grupoetario":"01","id_condicaootrab":"1","id_nivelescolaridade":"55","valor":0.000000000000000e+
000}, {"id_dado":28,"id_ano":"57A2011","id_localre":"1","id_sexo":"1","id_grupoetario":"01","id_condicaootrab":"1","id_nivelescolaridade":"17","valor":2.241000000000000e+
004}, {"id_dado":29,"id_ano":"57A2011","id_localre":"1","id_sexo":"1","id_grupoetario":"01","id_condicaootrab":"1","id_nivelescolaridade":"1","valor":1.940000000000000e+
002}, {"id_dado":30,"id_ano":"57A2011","id_localre":"1","id_sexo":"1","id_grupoetario":"01","id_condicaootrab":"12","id_nivelescolaridade":"2","valor":1.305900000000000e+
004}, {"id_dado":31,"id_ano":"57A2011","id_localre":"1","id_sexo":"1","id_grupoetario":"01","id_condicaootrab":"12","id_nivelescolaridade":"21","valor":6.520000000000000e+
002}, {"id_dado":32,"id_ano":"57A2011","id_localre":"1","id_sexo":"1","id_grupoetario":"01","id_condicaootrab":"12","id_nivelescolaridade":"22","valor":3.254000000000000e+
003}, {"id_dado":33,"id_ano":"57A2011","id_localre":"1","id_sexo":"1","id_grupoetario":"01","id_condicaootrab":"12","id_nivelescolaridade":"23","valor":9.073000000000000e+
003}, {"id_dado":34,"id_ano":"57A2011","id_localre":"1","id_sexo":"1","id_grupoetario":"01","id_condicaootrab":"12","id_nivelescolaridade":"3","valor":3.139000000000000e+
003}

```

FIGURA 4.5: Visualização dos dados no formato json

4.4 Procedimento que deverá validar e retornar uma informação



```
SQLQuery1.sql - DES...T3.master (sa (68))* X
CREATE PROCEDURE ValidatePermissions
AS
BEGIN
    DECLARE @Result NVARCHAR(100);
    DECLARE @HasPermissions BIT;

    IF IS_MEMBER('db_owner') = 1 OR IS_SRVROLEMEMBER('sysadmin') = 1
    BEGIN
        SET @HasPermissions = 1;
    END
    ELSE
    BEGIN
        SET @HasPermissions = 0;
    END

    IF @HasPermissions = 1
    BEGIN
        SET @Result = 'Você tem as permissões necessárias';
    END
    ELSE
    BEGIN
        SET @Result = 'Não tem credenciais / permissões';
    END

    SELECT @Result AS Resultado;
END;
GO
```

FIGURA 4.6: Procedimento para verificar permissões

O procedimento verifica se o utilizador que o executa possui permissões de administrador da base de dados ou do servidor. Ele declara variáveis para armazenar o resultado e as permissões. Depois verifica se o utilizador é membro dos grupos db owner ou sysadmin, e define uma mensagem de resultado baseada nessa verificação. Se o utilizador tiver as permissões necessárias, a mensagem é "Você tem as permissões necessárias", caso contrário, a mensagem é "Não tem credenciais/permissões". Finalmente, o procedimento retorna a mensagem de resultado ao utilizador.

4.5 Procedimento que insira numa tabela baseado num dataset no formato JSON

```
SQLQuery1.sql - DE...BD_censos (sa (73))" -> X
CREATE PROCEDURE InserirDadosJSON
    @Json NVARCHAR(MAX)
AS
BEGIN
    SET NOCOUNT ON;

    CREATE TABLE #TempDados (
        id_ano NVARCHAR(10),
        valor INT,
        id_localre NVARCHAR(10),
        id_sexo NVARCHAR(10),
        id_grupoetario NVARCHAR(10),
        id_condicaotrab NVARCHAR(10),
        id_nivelescolaridade NVARCHAR(10)
    );

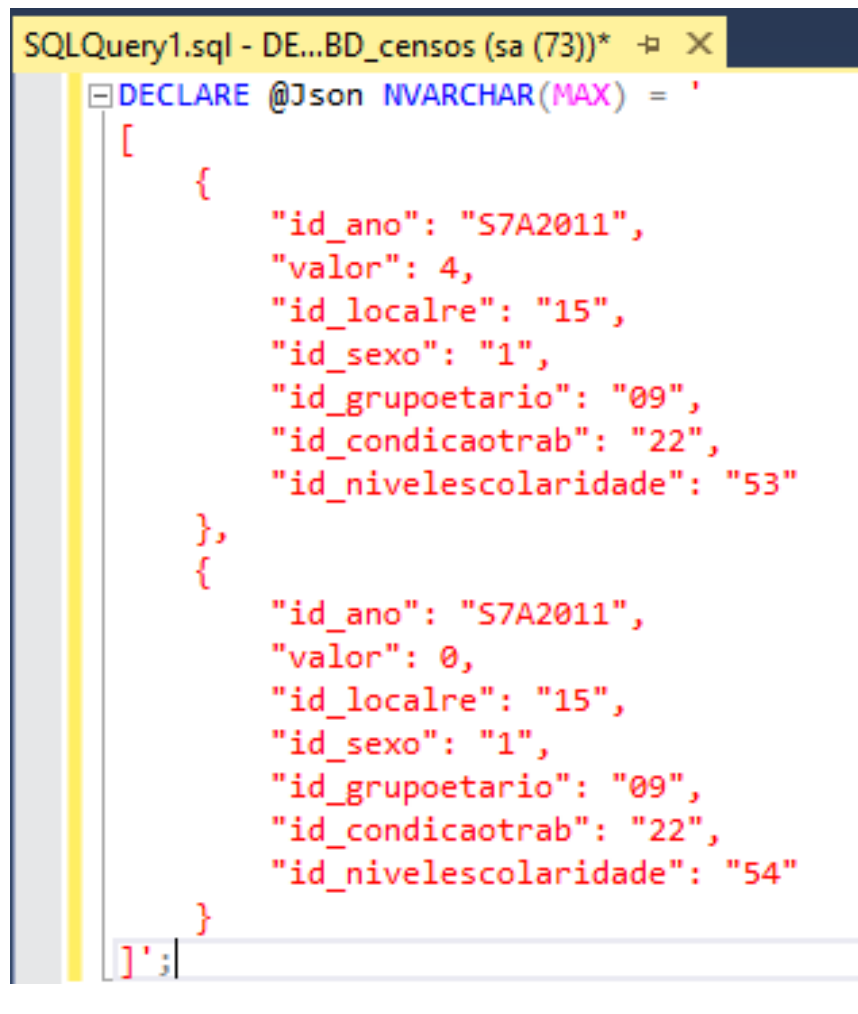
    INSERT INTO #TempDados (id_ano, valor, id_localre, id_sexo, id_grupoetario, id_condicaotrab, id_nivelescolaridade)
    SELECT
        JSON_VALUE(value, '$.id_ano') AS id_ano,
        JSON_VALUE(value, '$.valor') AS valor,
        JSON_VALUE(value, '$.id_localre') AS id_localre,
        JSON_VALUE(value, '$.id_sexo') AS id_sexo,
        JSON_VALUE(value, '$.id_grupoetario') AS id_grupoetario,
        JSON_VALUE(value, '$.id_condicaotrab') AS id_condicaotrab,
        JSON_VALUE(value, '$.id_nivelescolaridade') AS id_nivelescolaridade
    FROM OPENJSON(@Json);

    INSERT INTO dbo.dados (id_ano, valor, id_localre, id_sexo, id_grupoetario, id_condicaotrab, id_nivelescolaridade)
    SELECT
        id_ano, valor, id_localre, id_sexo, id_grupoetario, id_condicaotrab, id_nivelescolaridade
    FROM #TempDados;

    DROP TABLE #TempDados;
END;
```

FIGURA 4.7: Procedimento para inserir tabela baseada em dataset JSON

O procedimento insere dados em formato JSON em uma tabela na base de dados. O procedimento começa por desativar mensagens de contagem e criar uma tabela temporária TempDados com colunas para armazenar os dados extraídos do JSON. Em seguida, usa a função OPENJSON para transformar o JSON em registros e a função JSON VALUE para extrair valores específicos de cada campo do JSON, inserindo-os na tabela temporária. Depois, os dados da tabela temporária são transferidos para a tabela final dbo.dados. Finalmente, a tabela temporária é descartada.

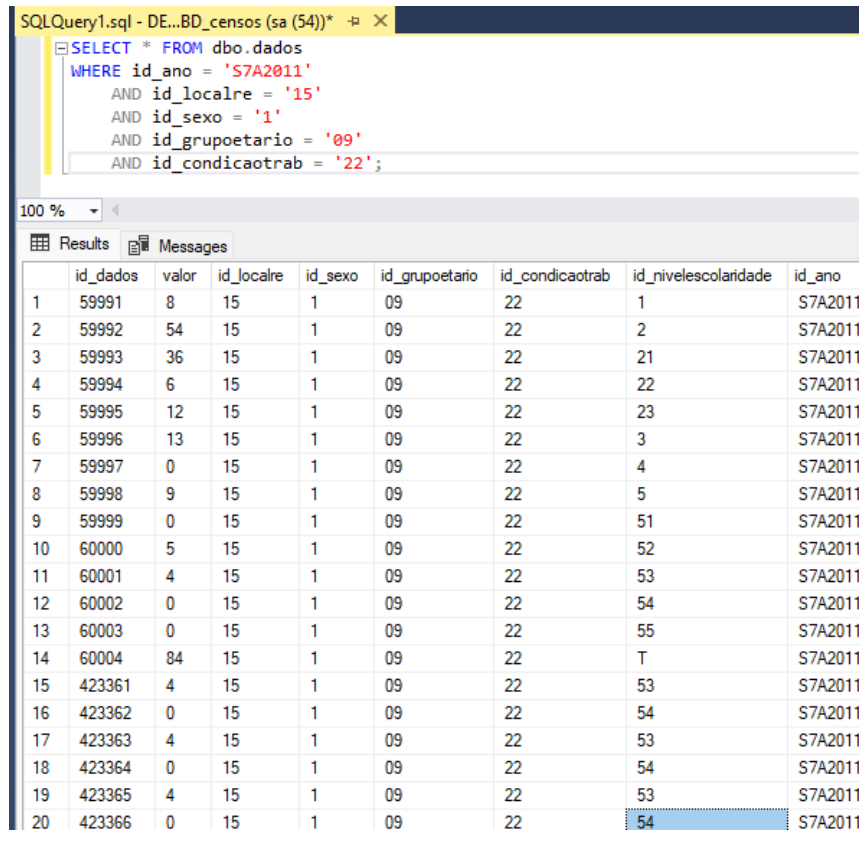
The image shows a screenshot of a SQL query editor window. The title bar reads "SQLQuery1.sql - DE...BD_censos (sa (73))*". The editor contains a SQL statement that declares a variable @Json of type NVARCHAR(MAX) and assigns it a JSON string. The JSON string is an array of two objects. Each object has seven fields: id_ano, valor, id_localre, id_sexo, id_grupoetario, id_condicaotrab, and id_nivelescolaridade. The first object has values: "S7A2011", 4, "15", "1", "09", "22", and "53". The second object has values: "S7A2011", 0, "15", "1", "09", "22", and "54". The code is as follows:

```
DECLARE @Json NVARCHAR(MAX) = '[
    {
        "id_ano": "S7A2011",
        "valor": 4,
        "id_localre": "15",
        "id_sexo": "1",
        "id_grupoetario": "09",
        "id_condicaotrab": "22",
        "id_nivelescolaridade": "53"
    },
    {
        "id_ano": "S7A2011",
        "valor": 0,
        "id_localre": "15",
        "id_sexo": "1",
        "id_grupoetario": "09",
        "id_condicaotrab": "22",
        "id_nivelescolaridade": "54"
    }
];'
```

FIGURA 4.8: Procedimento para inserir tabela baseada em dataset JSON

O código SQL declara uma variável @Json do tipo NVARCHAR(MAX) e a inicializa com uma string JSON contendo uma lista de dois objetos. Cada objeto possui campos como id ano, valor, id localre, id sexo, id grupoetario, id condicaotrab, e id nivelescolaridade, com valores específicos para cada campo. Essa variável JSON pode ser utilizada em operações subsequentes para manipulação ou inserção de dados na base de dados.

4.5. Procedimento que insira numa tabela baseado num dataset no formato JSON



The screenshot displays a SQL query window with the following text:

```
SELECT * FROM dbo.dados
WHERE id_ano = 'S7A2011'
AND id_localre = '15'
AND id_sexo = '1'
AND id_grupoetario = '09'
AND id_condicaotrab = '22';
```

Below the query, the 'Results' tab shows a table with 20 rows and 8 columns. The columns are: id_dados, valor, id_localre, id_sexo, id_grupoetario, id_condicaotrab, id_nivelescolaridade, and id_ano. The data is as follows:

	id_dados	valor	id_localre	id_sexo	id_grupoetario	id_condicaotrab	id_nivelescolaridade	id_ano
1	59991	8	15	1	09	22	1	S7A2011
2	59992	54	15	1	09	22	2	S7A2011
3	59993	36	15	1	09	22	21	S7A2011
4	59994	6	15	1	09	22	22	S7A2011
5	59995	12	15	1	09	22	23	S7A2011
6	59996	13	15	1	09	22	3	S7A2011
7	59997	0	15	1	09	22	4	S7A2011
8	59998	9	15	1	09	22	5	S7A2011
9	59999	0	15	1	09	22	51	S7A2011
10	60000	5	15	1	09	22	52	S7A2011
11	60001	4	15	1	09	22	53	S7A2011
12	60002	0	15	1	09	22	54	S7A2011
13	60003	0	15	1	09	22	55	S7A2011
14	60004	84	15	1	09	22	T	S7A2011
15	423361	4	15	1	09	22	53	S7A2011
16	423362	0	15	1	09	22	54	S7A2011
17	423363	4	15	1	09	22	53	S7A2011
18	423364	0	15	1	09	22	54	S7A2011
19	423365	4	15	1	09	22	53	S7A2011
20	423366	0	15	1	09	22	54	S7A2011

FIGURA 4.9: Resultado do procedimento

Capítulo 5

Cursosres

5.1 Cursor read only

```

DECLARE cursorread CURSOR READ_ONLY FOR
SELECT
  d.id_dado,
  d.valor,
  lr.descricao AS local_residencia,
  s.descricao AS sexo,
  ge.descricao AS grupo_etario,
  ct.descricao AS condicao_trabalho,
  ne.descricao AS nivel_escolaridade,
  a.ano
FROM
  dados d
INNER JOIN ano a ON d.id_ano = a.id_ano
INNER JOIN localresidencia lr ON d.id_localre = lr.id_localre
INNER JOIN sexo s ON d.id_sexo = s.id_sexo
INNER JOIN grupopetario ge ON d.id_grupoetario = ge.id_grupoetario
INNER JOIN condicaotrab ct ON d.id_condicaotrab = ct.id_condicaotrab
INNER JOIN nivelescolaridade ne ON d.id_nivelescolaridade = ne.id_nivelescolaridade
WHERE
  a.ano = 2011;

DECLARE @id_dado INT,
        @valor FLOAT,
        @local_residencia NVARCHAR(100),
        @sexo NVARCHAR(50),
        @grupo_etario NVARCHAR(50),
        @condicao_trabalho NVARCHAR(100),
        @nivel_escolaridade NVARCHAR(100),
        @ano INT;

-- Abrir o cursor
OPEN cursorread;

FETCH NEXT FROM cursorread INTO
  @id_dado,
  @valor,
  @local_residencia,
  @sexo,
  @grupo_etario,
  @condicao_trabalho,
  @nivel_escolaridade,
  @ano;

WHILE @@FETCH_STATUS = 0
BEGIN
  PRINT 'Dados ID: ' + CAST(@id_dado AS NVARCHAR(10)) +
    ', valor: ' + CAST(@valor AS NVARCHAR(10)) +
    ', residência: ' + @local_residencia +
    ', sexo: ' + @sexo +
    ', Grupo Etário: ' + @grupo_etario +
    ', Condição de Trabalho: ' + @condicao_trabalho +
    ', Nível Escolaridade: ' + @nivel_escolaridade +
    ', Ano: ' + CAST(@ano AS NVARCHAR(10));

  FETCH NEXT FROM cursorread INTO
    @id_dado,
    @valor,
    @local_residencia,
    @sexo,
    @grupo_etario,
    @condicao_trabalho,
    @nivel_escolaridade,
    @ano;
END

CLOSE cursorread;
DEALLOCATE cursorread;

```

FIGURA 5.1: Código do cursor para visualizar os registos efetuados em 2011

```

Dados ID: 1, Valor: 374, Residência: Continente, Sexo: M, Grupo Etário: 15 - 19 anos, Condição de Trabalho: Populacao Ativa, Nível Escolaridade: Nenhum, Ano: 2011
Dados ID: 2, Valor: 2987, Residência: Continente, Sexo: M, Grupo Etário: 15 - 19 anos, Condição de Trabalho: Populacao Ativa, Nível Escolaridade: Ensino basico, Ano: 2011
Dados ID: 3, Valor: 1219, Residência: Continente, Sexo: M, Grupo Etário: 15 - 19 anos, Condição de Trabalho: Populacao Ativa, Nível Escolaridade: 1 Ciclo, Ano: 2011
Dados ID: 4, Valor: 666, Residência: Continente, Sexo: M, Grupo Etário: 15 - 19 anos, Condição de Trabalho: Populacao Ativa, Nível Escolaridade: 2 Ciclos, Ano: 2011
Dados ID: 5, Valor: 2162, Residência: Continente, Sexo: M, Grupo Etário: 15 - 19 anos, Condição de Trabalho: Populacao Ativa, Nível Escolaridade: 3 Ciclos, Ano: 2011
Dados ID: 6, Valor: 5648, Residência: Continente, Sexo: M, Grupo Etário: 15 - 19 anos, Condição de Trabalho: Populacao Ativa, Nível Escolaridade: Ensino secundario, Ano: 2011
Dados ID: 7, Valor: 519, Residência: Continente, Sexo: M, Grupo Etário: 15 - 19 anos, Condição de Trabalho: Populacao Ativa, Nível Escolaridade: Ensino pos-secundario, Ano: 2011
Dados ID: 8, Valor: 1, Residência: Continente, Sexo: M, Grupo Etário: 15 - 19 anos, Condição de Trabalho: Populacao Ativa, Nível Escolaridade: Ensino superior, Ano: 2011
Dados ID: 9, Valor: 1, Residência: Continente, Sexo: M, Grupo Etário: 15 - 19 anos, Condição de Trabalho: Populacao Ativa, Nível Escolaridade: Curso tecnico superior profissional, Ano: 2011
Dados ID: 10, Valor: 180, Residência: Continente, Sexo: M, Grupo Etário: 15 - 19 anos, Condição de Trabalho: Empregados, Nível Escolaridade: Nenhum, Ano: 2011
Dados ID: 11, Valor: 0, Residência: Continente, Sexo: M, Grupo Etário: 15 - 19 anos, Condição de Trabalho: Populacao Ativa, Nível Escolaridade: Licenciatura, Ano: 2011
Dados ID: 12, Valor: 0, Residência: Continente, Sexo: M, Grupo Etário: 15 - 19 anos, Condição de Trabalho: Populacao Ativa, Nível Escolaridade: Mestrado, Ano: 2011
Dados ID: 13, Valor: 0, Residência: Continente, Sexo: M, Grupo Etário: 15 - 19 anos, Condição de Trabalho: Populacao Ativa, Nível Escolaridade: Doutoramento, Ano: 2011
Dados ID: 14, Valor: 19029, Residência: Continente, Sexo: M, Grupo Etário: 15 - 19 anos, Condição de Trabalho: Populacao Ativa, Nível Escolaridade: Total, Ano: 2011
Dados ID: 15, Valor: 1859, Residência: Continente, Sexo: M, Grupo Etário: 15 - 19 anos, Condição de Trabalho: Empregados, Nível Escolaridade: Ensino secundario, Ano: 2011
Dados ID: 16, Valor: 16429, Residência: Continente, Sexo: M, Grupo Etário: 15 - 19 anos, Condição de Trabalho: Empregados, Nível Escolaridade: Ensino basico, Ano: 2011
Dados ID: 17, Valor: 167, Residência: Continente, Sexo: M, Grupo Etário: 15 - 19 anos, Condição de Trabalho: Empregados, Nível Escolaridade: 1 Ciclo, Ano: 2011
Dados ID: 18, Valor: 1662, Residência: Continente, Sexo: M, Grupo Etário: 15 - 19 anos, Condição de Trabalho: Empregados, Nível Escolaridade: 2 Ciclos, Ano: 2011
Dados ID: 19, Valor: 1859, Residência: Continente, Sexo: M, Grupo Etário: 15 - 19 anos, Condição de Trabalho: Empregados, Nível Escolaridade: 3 Ciclos, Ano: 2011
Dados ID: 20, Valor: 1859, Residência: Continente, Sexo: M, Grupo Etário: 15 - 19 anos, Condição de Trabalho: Empregados, Nível Escolaridade: Ensino secundario, Ano: 2011
Dados ID: 21, Valor: 123, Residência: Continente, Sexo: M, Grupo Etário: 15 - 19 anos, Condição de Trabalho: Empregados, Nível Escolaridade: Ensino pos-secundario, Ano: 2011
Dados ID: 22, Valor: 1, Residência: Continente, Sexo: M, Grupo Etário: 15 - 19 anos, Condição de Trabalho: Empregados, Nível Escolaridade: Ensino superior, Ano: 2011
Dados ID: 23, Valor: 0, Residência: Continente, Sexo: M, Grupo Etário: 15 - 19 anos, Condição de Trabalho: Empregados, Nível Escolaridade: Curso tecnico superior profissional, Ano: 2011
Dados ID: 24, Valor: 1, Residência: Continente, Sexo: M, Grupo Etário: 15 - 19 anos, Condição de Trabalho: Empregados, Nível Escolaridade: Bacharelato, Ano: 2011
Dados ID: 25, Valor: 0, Residência: Continente, Sexo: M, Grupo Etário: 15 - 19 anos, Condição de Trabalho: Empregados, Nível Escolaridade: Licenciatura, Ano: 2011
Dados ID: 26, Valor: 0, Residência: Continente, Sexo: M, Grupo Etário: 15 - 19 anos, Condição de Trabalho: Empregados, Nível Escolaridade: Mestrado, Ano: 2011
Dados ID: 27, Valor: 0, Residência: Continente, Sexo: M, Grupo Etário: 15 - 19 anos, Condição de Trabalho: Empregados, Nível Escolaridade: Doutoramento, Ano: 2011
Dados ID: 28, Valor: 24441, Residência: Continente, Sexo: M, Grupo Etário: 15 - 19 anos, Condição de Trabalho: Empregados, Nível Escolaridade: Total, Ano: 2011
Dados ID: 29, Valor: 194, Residência: Continente, Sexo: M, Grupo Etário: 15 - 19 anos, Condição de Trabalho: Desempregados, Nível Escolaridade: Nenhum, Ano: 2011
Dados ID: 30, Valor: 1859, Residência: Continente, Sexo: M, Grupo Etário: 15 - 19 anos, Condição de Trabalho: Desempregados, Nível Escolaridade: Ensino basico, Ano: 2011
Dados ID: 31, Valor: 662, Residência: Continente, Sexo: M, Grupo Etário: 15 - 19 anos, Condição de Trabalho: Desempregados, Nível Escolaridade: 1 Ciclo, Ano: 2011
Dados ID: 32, Valor: 1859, Residência: Continente, Sexo: M, Grupo Etário: 15 - 19 anos, Condição de Trabalho: Desempregados, Nível Escolaridade: 2 Ciclos, Ano: 2011
Dados ID: 33, Valor: 973, Residência: Continente, Sexo: M, Grupo Etário: 15 - 19 anos, Condição de Trabalho: Desempregados, Nível Escolaridade: 3 Ciclos, Ano: 2011

```

FIGURA 5.2: Visualizar registos efetuados em 2011 a partir do cursor

```

DECLARE cursoranalise CURSOR read_only FOR
SELECT TOP 1 a.ano,
       s.descricao AS Sexo,
       ct.descricao AS CondicaoTrab,
       AVG(d.valor) AS MediaValor,
       COUNT(DISTINCT ge.id_grupoetario) AS ContagemGruposEtarios,
       COUNT(d.valor) AS TotalRegistros
FROM dados d
INNER JOIN sexo s ON d.id_sexo = s.id_sexo
INNER JOIN grupoetario ge ON d.id_grupoetario = ge.id_grupoetario
INNER JOIN condicaotrab ct ON d.id_condicaotrab = ct.id_condicaotrab
INNER JOIN ano a ON d.id_ano = a.id_ano
GROUP BY a.ano, s.descricao, ct.descricao
ORDER BY a.ano DESC, s.descricao

DECLARE @ano INT,
        @sexo VARCHAR(50),
        @condicaotrab VARCHAR(100),
        @mediavalor FLOAT,
        @contagemgruposetarios INT,
        @totalregistros INT

OPEN cursoranalise

FETCH NEXT FROM cursoranalise INTO @ano, @sexo, @condicaotrab, @mediavalor, @contagemgruposetarios, @totalregistros

WHILE @@FETCH_STATUS = 0
BEGIN
    PRINT 'Ano: ' + CAST(@ano AS VARCHAR) +
          ', Sexo: ' + @sexo +
          ', Condição de Trabalho: ' + @condicaotrab +
          ', Média de Valores: ' + CAST(@mediavalor AS VARCHAR) +
          ', Contagem de Grupos Etários: ' + CAST(@contagemgruposetarios AS VARCHAR) +
          ', Total de Registros: ' + CAST(@totalregistros AS VARCHAR)

    FETCH NEXT FROM cursoranalise INTO @ano, @sexo, @condicaotrab, @mediavalor, @contagemgruposetarios, @totalregistros
END

CLOSE cursoranalise
DEALLOCATE cursoranalise

```

FIGURA 5.3: Código da criação do cursor onde utiliza media e contagem dos valores dos registos.

Ano: 2021, Sexo: H, Condição de Trabalho: Desempregado, Média de Valores: 534.561, Contagem de Grupos Etários: 14, Total de Registros: 6960
 Completion time: 2024-06-01T22:24:09.980800+01:00

FIGURA 5.4: Visualizar o ano mais recente dos dados, media do atributo valor, numero de grupos etarios presentes nos registos e a quantidade dos valores.

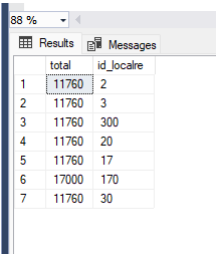
5.2 Cursor de Update

Foi observado que existem locais de residencia que se repetem mas com id diferentes, Área Metropolitana de Lisboa, Região Autónoma da Madeira e Região Autónoma dos Açores.

id_localre	descricao
12	150 Algarve
13	16 Centro
14	16B Oeste
15	16D Região de Aveiro
16	16E Região de Coimbra
17	16F Região de Leiria
18	16G Viseu Dão Lafões
19	16H Beira Baixa
20	16I Médio Tejo
21	16J Beiras e Serra da Estrela
22	17 Área Metropolitana de Lisboa
23	170 Área Metropolitana de Lisboa
24	18 Alentejo
25	181 Alentejo Litoral
26	184 Baixo Alentejo
27	185 Lezíria do Tejo
28	186 Alto Alentejo
29	187 Alentejo Central
30	2 Região Autónoma dos Açor...
31	20 Região Autónoma dos Açor...
32	3 Região Autónoma da Made...
33	30 Região Autónoma da Made...
34	300 Região Autónoma da Made...
35	PT Portugal

FIGURA 5.5: Locais de residência repetidos

Para corrigir este problema de ambiguidade foi desenvolvido um cursor que percorre todos os registos da tabela dados e no caso do registo ter um id de local de residência que é repetido é substituído pelo id original.



	total	id_localre
1	11760	2
2	11760	3
3	11760	300
4	11760	20
5	11760	17
6	17000	170
7	11760	30

FIGURA 5.6: Número de registos com locais de residência repetidos

```
DECLARE CursorUpdateLocalre CURSOR FOR
SELECT id_dados, id_localre
FROM dbo.dados;

DECLARE @id_dados INT;
DECLARE @id_localre VARCHAR(50);

OPEN CursorUpdateLocalre;

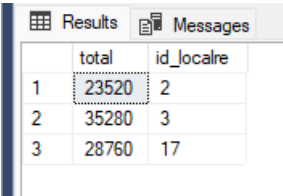
FETCH NEXT FROM CursorUpdateLocalre INTO @id_dados, @id_localre;

WHILE @@FETCH_STATUS = 0
BEGIN
    UPDATE dbo.dados
    SET id_localre =
        CASE
            WHEN @id_localre = '30' OR @id_localre = '300' THEN '3'
            WHEN @id_localre = '20' THEN '2'
            WHEN @id_localre = '170' THEN '17'
            ELSE id_localre
        END
    WHERE id_dados = @id_dados;

    FETCH NEXT FROM CursorUpdateLocalre INTO @id_dados, @id_localre;
END;

CLOSE CursorUpdateLocalre;
DEALLOCATE CursorUpdateLocalre;
```

FIGURA 5.7: Código do procedimento para gerir dados das tabelas de referência



	total	id_localre
1	23520	2
2	35280	3
3	28760	17

FIGURA 5.8: Verificar resultado depois da execução do cursor

Capítulo 6

Triggers

6.1 Trigger de validação

Nem todos os anos são anos de censos, para isso foi desenvolvido um trigger que antes da inserção na tabela anos valida se o ano é um válido de censos em Portugal. Anos de censos são de 10 em 10 anos (exemplos: 1991, 2001, 2011, 2021) para fazer esta validação se o resto da divisão do ano por 10 é diferente de 1 significa que o ano não é ano de censo e consequentemente não pode ser inserido nos registos.

```

CREATE TRIGGER validar_ano_censos
ON dbo.anos
INSTEAD OF INSERT
AS
BEGIN
    DECLARE @id_ano VARCHAR(50);
    DECLARE @ano INT;
    SELECT @id_ano = id_ano, @ano = ano FROM inserted;

    IF @ano % 10 != 1
    BEGIN
        RAISERROR('Ano inválido. O ano deve ser um ano de censo (por exemplo, 2001, 2011, 2021, etc.).', 16, 1);
    END
    ELSE
    BEGIN
        INSERT INTO dbo.anos(id_ano, ano) VALUES(@id_ano, @ano)
    END
END;

```

FIGURA 6.1: Código do trigger de validação antes da inserção

```

Msg 50000, Level 16, State 1, Procedure validar_ano_censos, Line 12 [Batch Start Line 0]
Ano inválido. O ano deve ser um ano de censo (por exemplo, 2001, 2011, 2021, etc.).

(1 row affected)

Completion time: 2024-05-30T14:28:27.7880188+01:00

```

FIGURA 6.2: Erro na inserção

```

(1 row affected)

(1 row affected)

Completion time: 2024-05-30T14:42:31.7149980+01:00

```

FIGURA 6.3: Inserção válida

6.2 Trigger com notificação por email

Para desenvolver um trigger com notificação por email é necessário começar por configurar o serviço Database Mail.

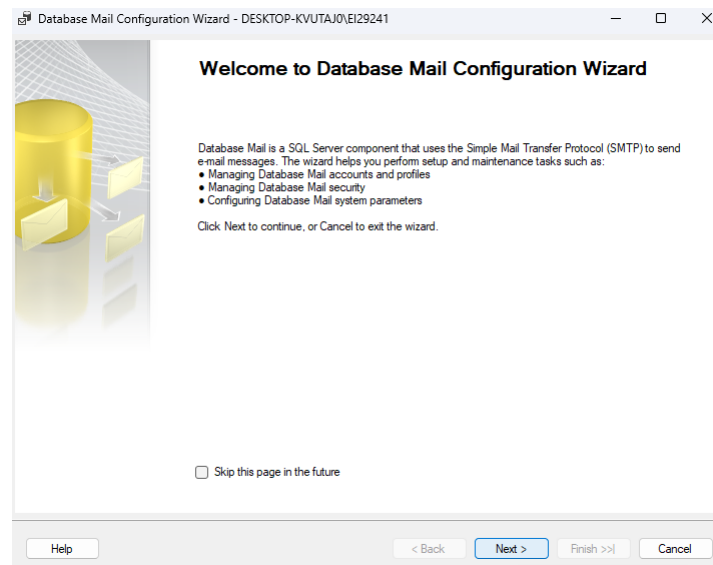


FIGURA 6.4: Na página Welcome to Database Mail Configuration Wizard clicar em next

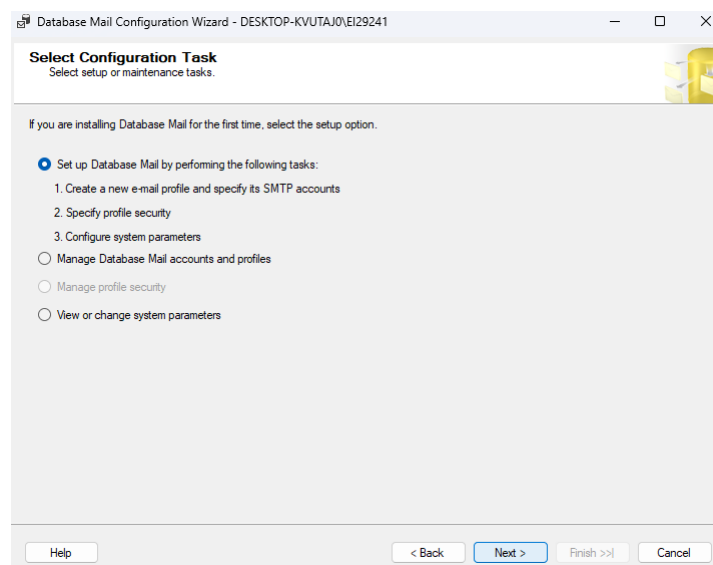


FIGURA 6.5: Na página Select Configuration Task selecionar Set up Database Mail by performing the following tasks e clicar em Next

Database Mail Configuration Wizard - DESKTOP-KVUTAJ0\EI29241

New Profile
Specify the profile name, description, accounts, and failover priority.

Profile name:

Description:

A profile may be associated with multiple SMTP accounts. If an account fails while sending an e-mail, the profile uses the next account in the priority list. Specify the accounts associated with the profile, and move the accounts to set the failover priority.

SMTP accounts:

Priority	Account Name	Email Address
----------	--------------	---------------

FIGURA 6.6: Na página New Profile preencher a caixa Profile name com o nome do perfil e clicar em Add

New Database Mail Account

Specify name, description, and attributes for your SMTP account.

Account name:

Description:

Outgoing Mail Server (SMTP)

Email address:

Display name:

Reply e-mail:

Server name: Port number:

☐ This server requires a secure connection (SSL)

SMTP Authentication

☐ Windows Authentication using Database Engine service credentials

☒ Basic authentication

User name:

Password:

Confirm password:

☐ Anonymous authentication

FIGURA 6.7: Na página New Database Mail Account preencher os dados da conta mail e clicar em Ok

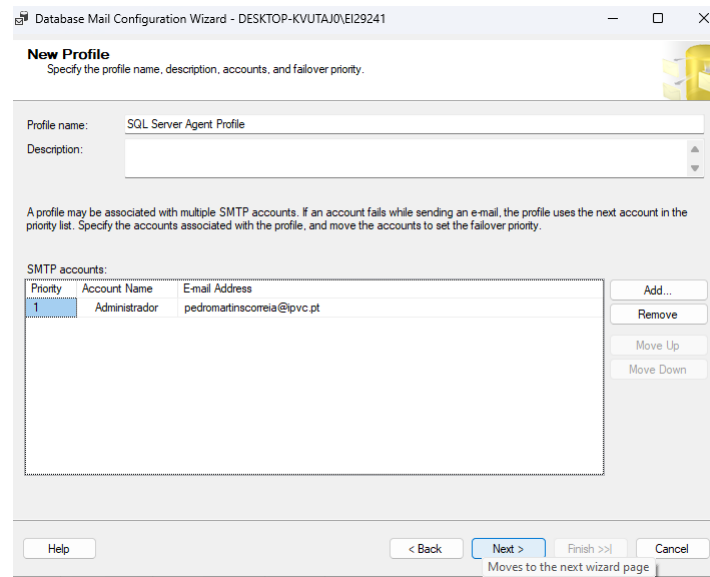


FIGURA 6.8: Na página New Profile clicar em Next

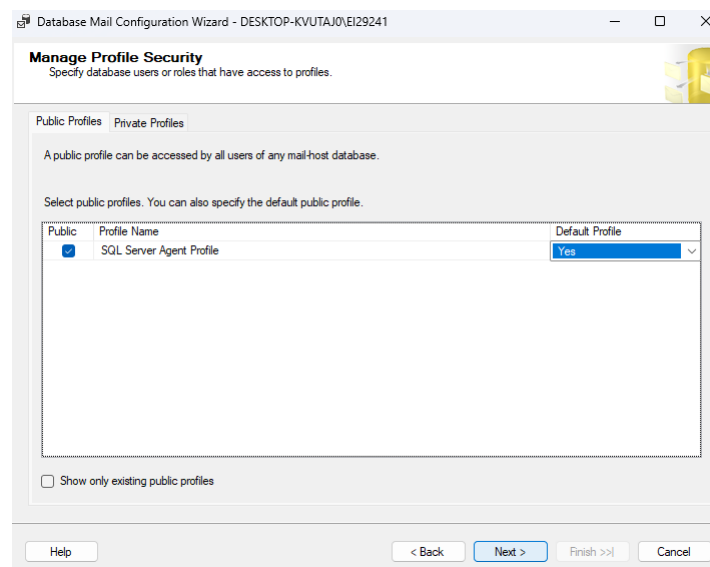


FIGURA 6.9: Na página Manage Profile Security seleccionar o perfil como Public, definir o Default Profile como Yes e clicar em Next

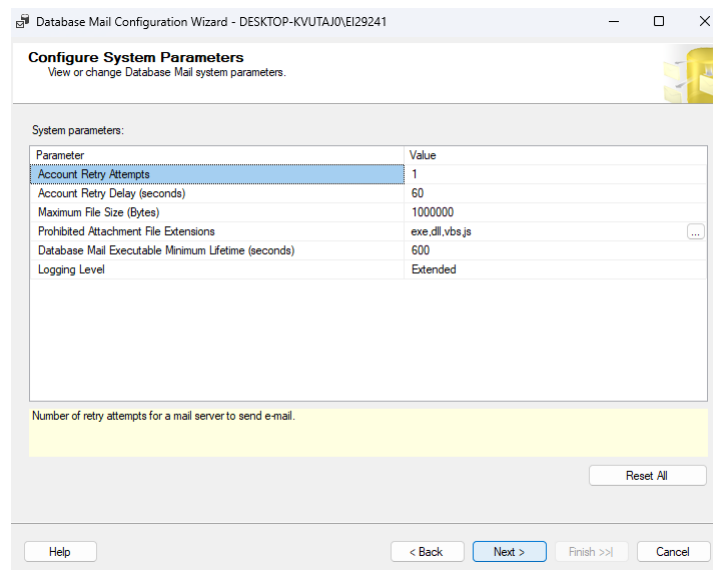


FIGURA 6.10: Na página Configure System Parameters clicar em Next

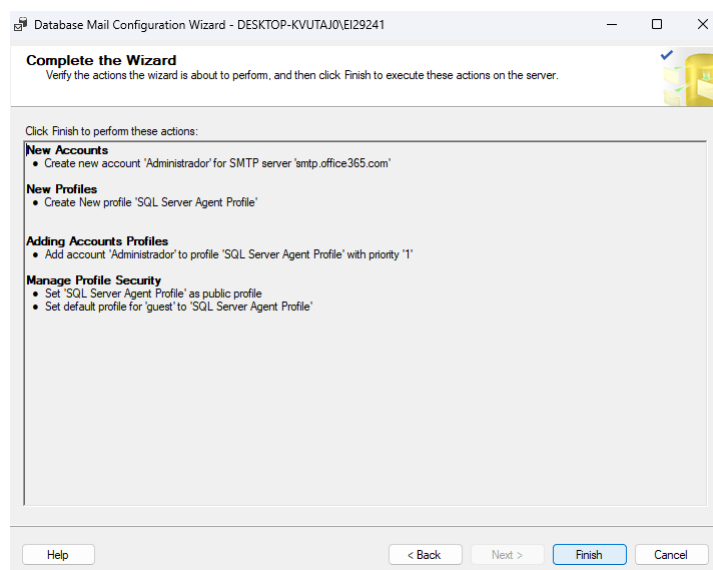


FIGURA 6.11: Na página Complete the Wizard clicar em Finish

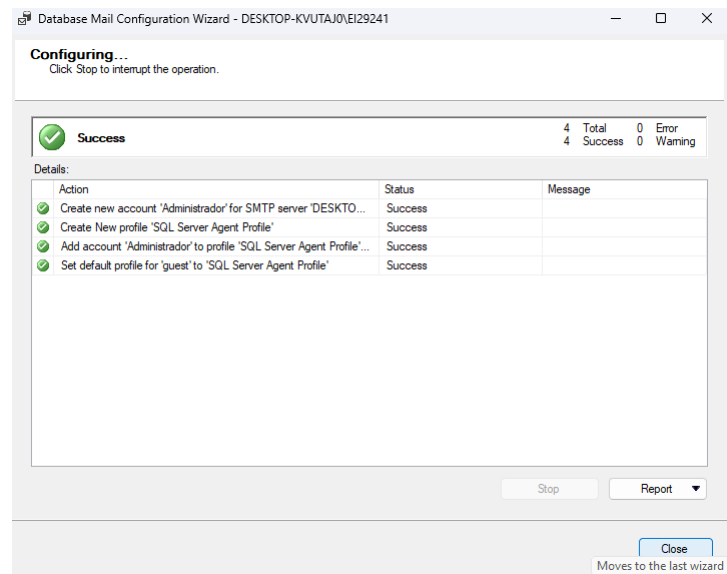


FIGURA 6.12: Quando a configuração estiver concluída clicar em Close

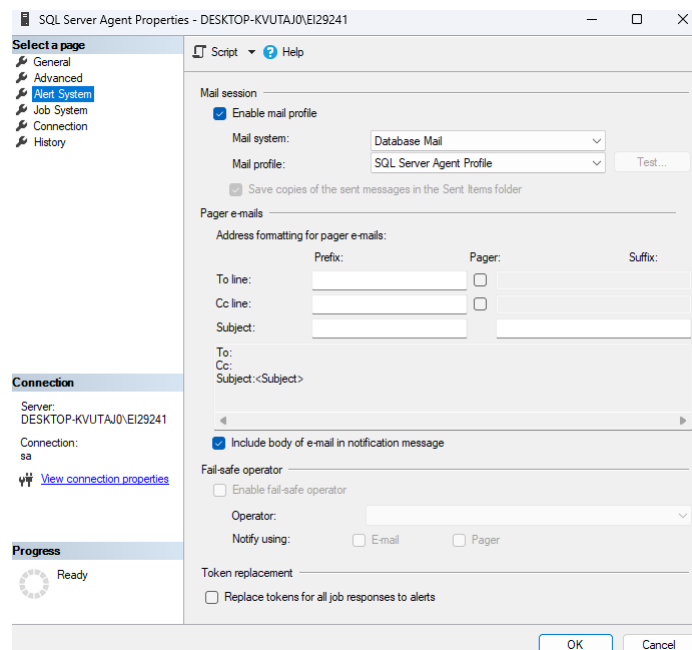


FIGURA 6.13: Na página SQL Server Agent Properties na aba Alert System ativar a opção Enable mail profile

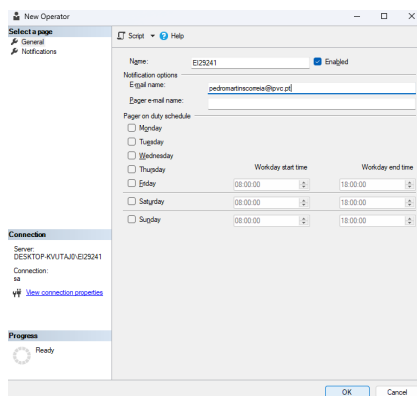


FIGURA 6.14: Na página New Operator preencher os devidos campos

O trigger desenvolvido tem como objetivo enviar um email quando um ano de censos tem um número elevado de registos. Neste caso o valor que define se é um número elevado ou não é 195807 uma vez que existiam 195806 no ano 2011, foi definido este valor para facilitar a confirmação da execução do trigger. Para que o trigger envie um email é necessário invocar o `sp_send_dbmail` e preencher os parametros do mesmo.

```
CREATE TRIGGER trigger_notificacao_email
ON dbo.dados
AFTER INSERT
AS
BEGIN
    DECLARE @id_ano VARCHAR(50);
    SELECT @id_ano = id_ano
    FROM inserted;

    IF (SELECT COUNT(*) FROM dados WHERE id_ano = @id_ano) = 195807
    BEGIN
        DECLARE @body NVARCHAR(MAX);
        SET @body = 'O ano ' + CAST(@id_ano AS NVARCHAR) + ' atingiu os 195807 registos.';

        EXEC msdb.dbo.sp_send_dbmail
            @profile_name = 'SQL Server Mail Profile',
            @recipients = 'pedromartinscorreia@ipvc.pt',
            @subject = 'Notificação de Contagem de Registos',
            @body = @body;
    END
END;
```

FIGURA 6.15: Código do trigger com notificação por email

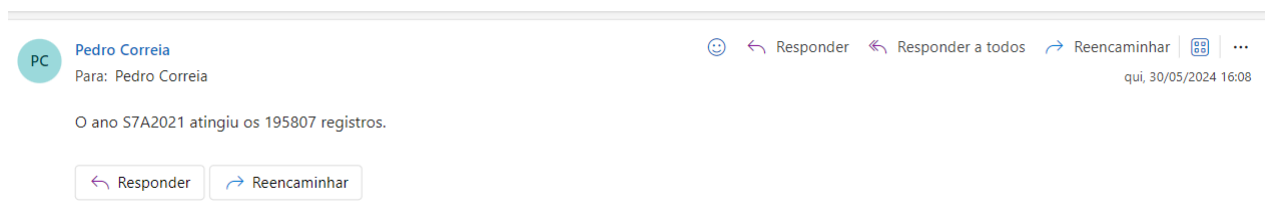


FIGURA 6.16: Email recebido depois do insert

6.3 Trigger que dispara após ser criado, alterado ou eliminado um registo.


```
CREATE TRIGGER tigger1
ON dados
AFTER INSERT, UPDATE, DELETE
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @tipoOperacao AS VARCHAR(10);

    IF EXISTS (SELECT * FROM inserted) AND EXISTS (SELECT * FROM deleted)
    BEGIN
        SET @tipoOperacao = 'UPDATE';
    END
    ELSE IF EXISTS (SELECT * FROM inserted)
    BEGIN
        SET @tipoOperacao = 'INSERT';
    END
    ELSE
    BEGIN
        SET @tipoOperacao = 'DELETE';
    END

    DECLARE @descricao AS VARCHAR(100);

    IF @tipoOperacao = 'INSERT'
    BEGIN
        SELECT @descricao = 'Inserido novo registro com ID: ' + CAST(id_dado AS VARCHAR)
        FROM inserted;
    END
    ELSE IF @tipoOperacao = 'UPDATE'
    BEGIN
        SELECT @descricao = 'Atualizado registro com ID: ' + CAST(id_dado AS VARCHAR)
        FROM inserted;
    END
    ELSE
    BEGIN
        SELECT @descricao = 'Deletado registro com ID: ' + CAST(id_dado AS VARCHAR)
        FROM deleted;
    END

    -- Inserir registro de log
    INSERT INTO log_atualizacoes (tipo_operacao, descricao, data_hora)
    VALUES (@tipoOperacao, @descricao, GETDATE());
END
```

FIGURA 6.17: Trigger onde registra o movimento na tabela `log_atualizacoes` após `insert/update/delete`.

```
CREATE TABLE log_atualizacoes (
    log_id INT PRIMARY KEY IDENTITY(1,1),
    tipo_operacao VARCHAR(10),
    descricao VARCHAR(100),
    data_hora DATETIME
);
```

FIGURA 6.18: Código para criar a tabela `logAtualizacoes`

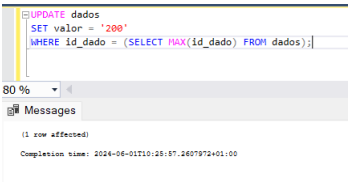


FIGURA 6.19: Código para testar o trigger a partir de um update

SQLQuery2.sql - LAP...abpratico (sa (57))

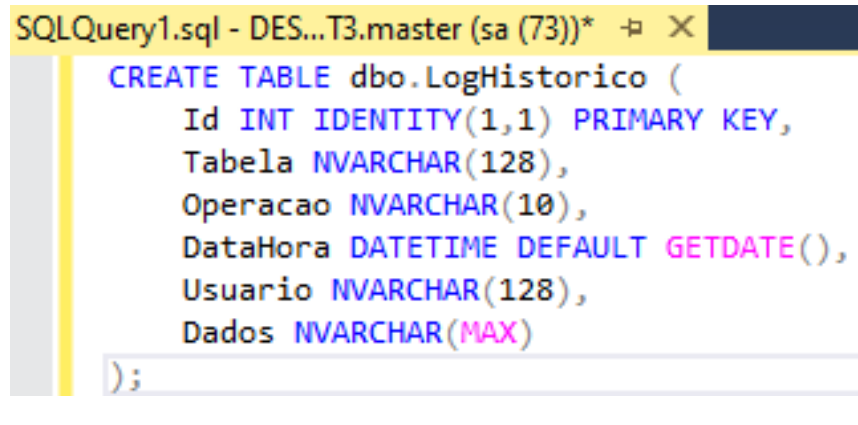
```
SELECT * FROM log_atualizacoes
```

log_id	tipo_operacao	descricao	data_hora
1	UPDATE	Atualizado registro com ID: 470338	2024-06-01 10:25:57.240
2	UPDATE	Atualizado registro com ID: 470338	2024-06-01 10:25:57.257

FIGURA 6.20: Visualização do movimento criado a partir do update na tabela `logAtualizacoes`

6.4 Tabela de logs de todas as interações da base de dados alimentada com triggers

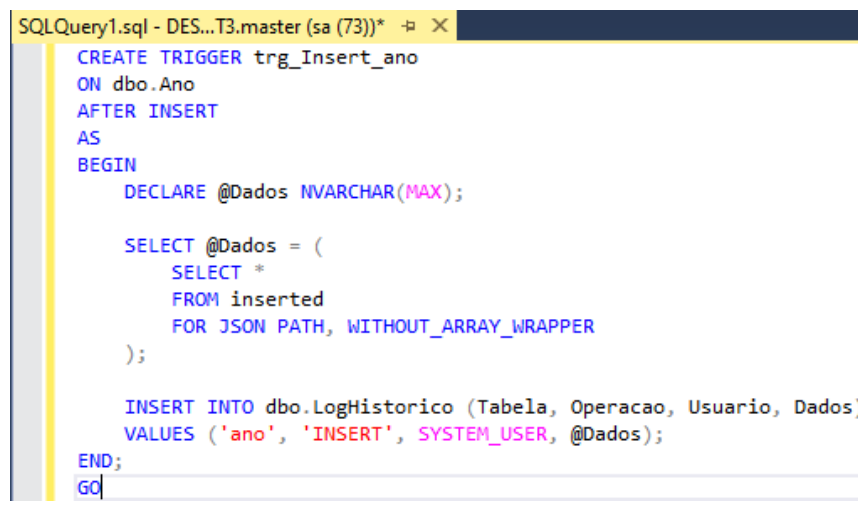
Este código cria a tabela LogHistorico que é usada para armazenar registros de operações realizadas em outras tabelas, incluindo detalhes como o nome da tabela, tipo de operação, data e hora da operação, utilizador que realizou a operação e os dados envolvidos.



```
SQLQuery1.sql - DES...T3.master (sa (73))* X
CREATE TABLE dbo.LogHistorico (
    Id INT IDENTITY(1,1) PRIMARY KEY,
    Tabela NVARCHAR(128),
    Operacao NVARCHAR(10),
    DataHora DATETIME DEFAULT GETDATE(),
    Usuario NVARCHAR(128),
    Dados NVARCHAR(MAX)
);
```

FIGURA 6.21: Criação da tabela LogHistorico

Agora vão ser criados três triggers para cada uma das tabelas já existentes. Um trigger para registrar automaticamente qualquer inserção. Outro para registrar automaticamente qualquer atualização nas tabelas. E outro para registrar qualquer eliminação das tabelas.



```
SQLQuery1.sql - DES...T3.master (sa (73))* X
CREATE TRIGGER trg_Insert_ano
ON dbo.Ano
AFTER INSERT
AS
BEGIN
    DECLARE @Dados NVARCHAR(MAX);

    SELECT @Dados = (
        SELECT *
        FROM inserted
        FOR JSON PATH, WITHOUT_ARRAY_WRAPPER
    );

    INSERT INTO dbo.LogHistorico (Tabela, Operacao, Usuario, Dados)
    VALUES ('ano', 'INSERT', SYSTEM_USER, @Dados);
END;
GO
```

FIGURA 6.22: Trigger para registrar qualquer inserção na tabela ano

```

SQLQuery1.sql - DES...T3.master (sa (73))* X
CREATE TRIGGER trg_Update_ano
ON dbo.Ano
AFTER UPDATE
AS
BEGIN
    DECLARE @Dados NVARCHAR(MAX);

    SELECT @Dados = (
        SELECT *
        FROM inserted
        FOR JSON PATH, WITHOUT_ARRAY_WRAPPER
    );

    INSERT INTO dbo.LogHistorico (Tabela, Operacao, Usuario, Dados)
    VALUES ('ano', 'UPDATE', SYSTEM_USER, @Dados);
END;
GO

```

FIGURA 6.23: Trigger para registrar qualquer atualização na tabela ano

```

SQLQuery1.sql - DES...T3.master (sa (73))* X
CREATE TRIGGER trg_Delete_ano
ON dbo.Ano
AFTER DELETE
AS
BEGIN
    DECLARE @Dados NVARCHAR(MAX);

    SELECT @Dados = (
        SELECT *
        FROM deleted
        FOR JSON PATH, WITHOUT_ARRAY_WRAPPER
    );

    INSERT INTO dbo.LogHistorico (Tabela, Operacao, Usuario, Dados)
    VALUES ('ano', 'DELETE', SYSTEM_USER, @Dados);
END;
GO

```

FIGURA 6.24: Trigger para registrar qualquer eliminação na tabela ano

```

SQLQuery1.sql - DES...T3.master (sa (73))* X
CREATE TRIGGER trg_Insert_sexo
ON dbo.sexo
AFTER INSERT
AS
BEGIN
    DECLARE @Dados NVARCHAR(MAX);
    SELECT @Dados = (SELECT * FROM inserted FOR JSON PATH, WITHOUT_ARRAY_WRAPPER);
    INSERT INTO dbo.LogHistorico (Tabela, Operacao, Usuario, Dados)
    VALUES ('sexo', 'INSERT', SYSTEM_USER, @Dados);
END;
GO

```

FIGURA 6.25: Trigger para registrar qualquer inserção na tabela sexo

```
SQLQuery1.sql - DES...T3.master (sa (73)) *  X
CREATE TRIGGER trg_Update_sexo
ON dbo.sexo
AFTER UPDATE
AS
BEGIN
    DECLARE @Dados NVARCHAR(MAX);
    SELECT @Dados = (SELECT * FROM inserted FOR JSON PATH, WITHOUT_ARRAY_WRAPPER);
    INSERT INTO dbo.LogHistorico (Tabela, Operacao, Usuario, Dados)
    VALUES ('sexo', 'UPDATE', SYSTEM_USER, @Dados);
END;
GO
```

FIGURA 6.26: Trigger para registrar qualquer atualização na tabela sexo

```
SQLQuery1.sql - DES...T3.master (sa (73)) *  X
CREATE TRIGGER trg_Delete_sexo
ON dbo.sexo
AFTER DELETE
AS
BEGIN
    DECLARE @Dados NVARCHAR(MAX);
    SELECT @Dados = (SELECT * FROM deleted FOR JSON PATH, WITHOUT_ARRAY_WRAPPER);
    INSERT INTO dbo.LogHistorico (Tabela, Operacao, Usuario, Dados)
    VALUES ('sexo', 'DELETE', SYSTEM_USER, @Dados);
END;
GO
```

FIGURA 6.27: Trigger para registrar qualquer inserção na tabela ano

```
SQLQuery1.sql - DES...T3.master (sa (73)) *  X
CREATE TRIGGER trg_Insert_condicaotrab
ON dbo.condicaotrab
AFTER INSERT
AS
BEGIN
    DECLARE @Dados NVARCHAR(MAX);
    SELECT @Dados = (SELECT * FROM inserted FOR JSON PATH, WITHOUT_ARRAY_WRAPPER);
    INSERT INTO dbo.LogHistorico (Tabela, Operacao, Usuario, Dados)
    VALUES ('condicaotrab', 'INSERT', SYSTEM_USER, @Dados);
END;
GO
```

FIGURA 6.28: Trigger para registrar qualquer inserção na tabela condição trabalho

```
SQLQuery1.sql - DES...T3.master (sa (73)) *  X
CREATE TRIGGER trg_Update_condicaotrab
ON dbo.condicaotrab
AFTER UPDATE
AS
BEGIN
    DECLARE @Dados NVARCHAR(MAX);
    SELECT @Dados = (SELECT * FROM inserted FOR JSON PATH, WITHOUT_ARRAY_WRAPPER);
    INSERT INTO dbo.LogHistorico (Tabela, Operacao, Usuario, Dados)
    VALUES ('condicaotrab', 'UPDATE', SYSTEM_USER, @Dados);
END;
GO
```

FIGURA 6.29: Trigger para registrar qualquer atualização na tabela condição trabalho

```
SQLQuery1.sql - DES...T3.master (sa (73)) *  X
CREATE TRIGGER trg_Delete_condicaotrab
ON dbo.condicaotrab
AFTER DELETE
AS
BEGIN
    DECLARE @Dados NVARCHAR(MAX);
    SELECT @Dados = (SELECT * FROM deleted FOR JSON PATH, WITHOUT_ARRAY_WRAPPER);
    INSERT INTO dbo.LogHistorico (Tabela, Operacao, Usuario, Dados)
    VALUES ('condicaotrab', 'DELETE', SYSTEM_USER, @Dados);
END;
GO
```

FIGURA 6.30: Trigger para registrar qualquer eliminação na tabela condição trabalho

```
SQLQuery1.sql - DES...T3.master (sa (73)) *  X
CREATE TRIGGER trg_Insert_nivelescolaridade
ON dbo.nivelescolaridade
AFTER INSERT
AS
BEGIN
    DECLARE @Dados NVARCHAR(MAX);
    SELECT @Dados = (SELECT * FROM inserted FOR JSON PATH, WITHOUT_ARRAY_WRAPPER);
    INSERT INTO dbo.LogHistorico (Tabela, Operacao, Usuario, Dados)
    VALUES ('nivelescolaridade', 'INSERT', SYSTEM_USER, @Dados);
END;
GO
```

FIGURA 6.31: Trigger para registrar qualquer inserção na tabela nível de escolaridade

```
SQLQuery1.sql - DES...T3.master (sa (73)) *  X
CREATE TRIGGER trg_Update_nivelescolaridade
ON dbo.nivelescolaridade
AFTER UPDATE
AS
BEGIN
    DECLARE @Dados NVARCHAR(MAX);
    SELECT @Dados = (SELECT * FROM inserted FOR JSON PATH, WITHOUT_ARRAY_WRAPPER);
    INSERT INTO dbo.LogHistorico (Tabela, Operacao, Usuario, Dados)
    VALUES ('nivelescolaridade', 'UPDATE', SYSTEM_USER, @Dados);
END;
GO
```

FIGURA 6.32: Trigger para registrar qualquer atualização na tabela nível de escolaridade

```
SQLQuery1.sql - DES...T3.master (sa (73)) * X
CREATE TRIGGER trg_Delete_nivelescolaridade
ON dbo.nivelescolaridade
AFTER DELETE
AS
BEGIN
    DECLARE @Dados NVARCHAR(MAX);
    SELECT @Dados = (SELECT * FROM deleted FOR JSON PATH, WITHOUT_ARRAY_WRAPPER);
    INSERT INTO dbo.LogHistorico (Tabela, Operacao, Usuario, Dados)
    VALUES ('nivelescolaridade', 'DELETE', SYSTEM_USER, @Dados);
END;
Go
```

FIGURA 6.33: Trigger para registrar qualquer eliminação na tabela nível de escolaridade

```
SQLQuery1.sql - DES...T3.master (sa (73)) * X
CREATE TRIGGER trg_Insert_grupoetario
ON dbo.grupoetario
AFTER INSERT
AS
BEGIN
    DECLARE @Dados NVARCHAR(MAX);
    SELECT @Dados = (SELECT * FROM inserted FOR JSON PATH, WITHOUT_ARRAY_WRAPPER);
    INSERT INTO dbo.LogHistorico (Tabela, Operacao, Usuario, Dados)
    VALUES ('grupoetario', 'INSERT', SYSTEM_USER, @Dados);
END;
Go
```

FIGURA 6.34: Trigger para registrar qualquer inserção na tabela grupo etário

```
SQLQuery1.sql - DES...T3.master (sa (73)) * X
CREATE TRIGGER trg_Update_grupoetario
ON dbo.grupoetario
AFTER UPDATE
AS
BEGIN
    DECLARE @Dados NVARCHAR(MAX);
    SELECT @Dados = (SELECT * FROM inserted FOR JSON PATH, WITHOUT_ARRAY_WRAPPER);
    INSERT INTO dbo.LogHistorico (Tabela, Operacao, Usuario, Dados)
    VALUES ('grupoetario', 'UPDATE', SYSTEM_USER, @Dados);
END;
Go
```

FIGURA 6.35: Trigger para registrar qualquer atualização na tabela grupo etário

```
SQLQuery1.sql - DES...T3.master (sa (73))* X
CREATE TRIGGER trg_Delete_grupoetario
ON dbo.grupoetario
AFTER DELETE
AS
BEGIN
    DECLARE @Dados NVARCHAR(MAX);
    SELECT @Dados = (SELECT * FROM deleted FOR JSON PATH, WITHOUT_ARRAY_WRAPPER);
    INSERT INTO dbo.LogHistorico (Tabela, Operacao, Usuario, Dados)
    VALUES ('grupoetario', 'DELETE', SYSTEM_USER, @Dados);
END;
GO
```

FIGURA 6.36: Trigger para registrar qualquer eliminação na tabela grupo etário

```
SQLQuery1.sql - DES...T3.master (sa (73))* X
CREATE TRIGGER trg_Insert_localresidencia
ON dbo.localresidencia
AFTER INSERT
AS
BEGIN
    DECLARE @Dados NVARCHAR(MAX);
    SELECT @Dados = (SELECT * FROM inserted FOR JSON PATH, WITHOUT_ARRAY_WRAPPER);
    INSERT INTO dbo.LogHistorico (Tabela, Operacao, Usuario, Dados)
    VALUES ('localresidencia', 'INSERT', SYSTEM_USER, @Dados);
END;
GO
```

FIGURA 6.37: Trigger para registrar qualquer inserção na tabela local de residência

```
SQLQuery1.sql - DES...T3.master (sa (73))* X
CREATE TRIGGER trg_Update_localresidencia
ON dbo.localresidencia
AFTER UPDATE
AS
BEGIN
    DECLARE @Dados NVARCHAR(MAX);
    SELECT @Dados = (SELECT * FROM inserted FOR JSON PATH, WITHOUT_ARRAY_WRAPPER);
    INSERT INTO dbo.LogHistorico (Tabela, Operacao, Usuario, Dados)
    VALUES ('localresidencia', 'UPDATE', SYSTEM_USER, @Dados);
END;
GO
```

FIGURA 6.38: Trigger para registrar qualquer atualização na tabela local de residência

```
SQLQuery1.sql - DES...T3.master (sa (73))* X
CREATE TRIGGER trg_Delete_localresidencia
ON dbo.localresidencia
AFTER DELETE
AS
BEGIN
    DECLARE @Dados NVARCHAR(MAX);
    SELECT @Dados = (SELECT * FROM deleted FOR JSON PATH, WITHOUT_ARRAY_WRAPPER);
    INSERT INTO dbo.LogHistorico (Tabela, Operacao, Usuario, Dados)
    VALUES ('localresidencia', 'DELETE', SYSTEM_USER, @Dados);
END;
GO
```

FIGURA 6.39: Trigger para registrar qualquer eliminação na tabela local de residência

```
SQLQuery1.sql - DES...T3.master (sa (73))* -p X
CREATE TRIGGER trg_Insert_dados
ON dbo.dados
AFTER INSERT
AS
BEGIN
    DECLARE @Dados NVARCHAR(MAX);
    SELECT @Dados = (SELECT * FROM inserted FOR JSON PATH, WITHOUT_ARRAY_WRAPPER);
    INSERT INTO dbo.LogHistorico (Tabela, Operacao, Usuario, Dados)
    VALUES ('dados', 'INSERT', SYSTEM_USER, @Dados);
END;
GO
```

FIGURA 6.40: Trigger para registrar qualquer inserção na tabela dados

```
SQLQuery1.sql - DE...BD_censos (sa (73))* -p X
CREATE TRIGGER trg_Update_dados
ON dbo.dados
AFTER UPDATE
AS
BEGIN
    DECLARE @Dados NVARCHAR(MAX);
    SELECT @Dados = (SELECT * FROM inserted FOR JSON PATH, WITHOUT_ARRAY_WRAPPER);
    INSERT INTO dbo.LogHistorico (Tabela, Operacao, Usuario, Dados)
    VALUES ('dados', 'UPDATE', SYSTEM_USER, @Dados);
END;
GO
```

FIGURA 6.41: Trigger para registrar qualquer atualização na tabela dados

```
SQLQuery1.sql - DE...BD_censos (sa (73))* -p X
CREATE TRIGGER trg_Delete_dados
ON dbo.dados
AFTER DELETE
AS
BEGIN
    DECLARE @Dados NVARCHAR(MAX);
    SELECT @Dados = (SELECT * FROM deleted FOR JSON PATH, WITHOUT_ARRAY_WRAPPER);
    INSERT INTO dbo.LogHistorico (Tabela, Operacao, Usuario, Dados)
    VALUES ('dados', 'DELETE', SYSTEM_USER, @Dados);
END;
GO
```

FIGURA 6.42: Trigger para registrar qualquer eliminação na tabela dados

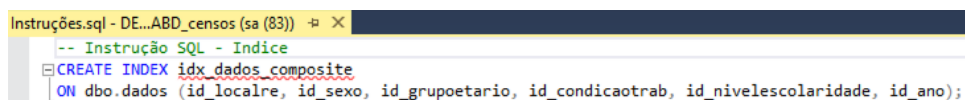
Esses triggers garantem que todas as operações importantes (inserção, atualização e eliminação) nas tabelas sejam registradas em uma tabela de histórico para auditoria e rastreamento de mudanças.

Capítulo 7

Instruções SQL

7.1 Criação de um Índice

Um índice em SQL é uma estrutura que melhora a velocidade das operações de consulta numa tabela de base de dados, embora possa diminuir a velocidade de operações de escrita, como inserções, atualizações e eliminações.



```

-- Instrução SQL - Índice
CREATE INDEX idx_dados_composite
ON dbo.dados (id_localre, id_sexo, id_grupoetario, id_condicaotrab, id_nivelescolaridade, id_ano);

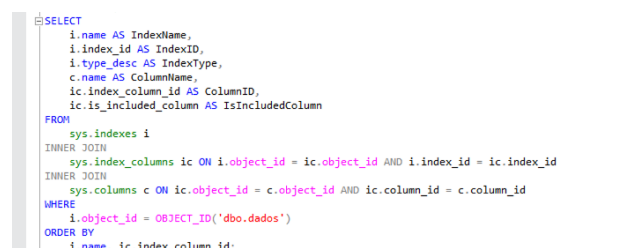
```

FIGURA 7.1: Código para criar índice

Este comando cria um índice chamado idx.dados.composite na tabela dbo.dados, envolvendo as seguintes colunas:

- id.localre
- id.sexo
- id.grupoetario
- id.condicoatrab
- id.nivelescolaridade
- id.ano

Um índice composto como este é utilizado para otimizar consultas que filtram ou ordenam dados com base numa combinação dessas colunas. É especialmente útil quando se consulta ou ordena frequentemente os dados com base nessas colunas em conjunto.

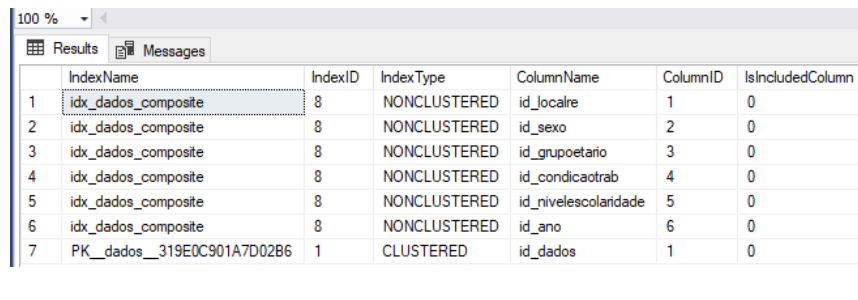


```

SELECT
    i.name AS IndexName,
    i.index_id AS IndexID,
    i.type_desc AS IndexType,
    c.name AS ColumnName,
    ic.index_column_id AS ColumnID,
    ic.is_included_column AS IsIncludedColumn
FROM
    sys.indexes i
    INNER JOIN
    sys.index_columns ic ON i.object_id = ic.object_id AND i.index_id = ic.index_id
    INNER JOIN
    sys.columns c ON ic.object_id = c.object_id AND ic.column_id = c.column_id
WHERE
    i.object_id = OBJECT_ID('dbo.dados')
ORDER BY
    i.name, ic.index_column_id;

```

FIGURA 7.2: Visualizar código de Índice



	IndexName	IndexID	IndexType	ColumnName	ColumnID	IsIncludedColumn
1	idx_dados_composite	8	NONCLUSTERED	id_localre	1	0
2	idx_dados_composite	8	NONCLUSTERED	id_sexo	2	0
3	idx_dados_composite	8	NONCLUSTERED	id_grupoetario	3	0
4	idx_dados_composite	8	NONCLUSTERED	id_condicaootrab	4	0
5	idx_dados_composite	8	NONCLUSTERED	id_nivelescolaridade	5	0
6	idx_dados_composite	8	NONCLUSTERED	id_ano	6	0
7	PK_dados__319E0C901A7D02B6	1	CLUSTERED	id_dados	1	0

FIGURA 7.3: Índice

Esta consulta é usada para verificar os detalhes dos índices existentes na tabela `dbo.dados`.

Tabela `sys.indexes` (i):

- Contém informações sobre os índices definidos em uma tabela;
- `i.name` (IndexName) fornece o nome do índice.
- `i.index.id` (IndexID) fornece o identificador do índice.
- `i.type.desc` (IndexType) descreve o tipo de índice (por exemplo, `clustered`, `non-clustered`).

Tabela `sys.index.columns` (ic):

- Contém informações sobre as colunas que fazem parte dos índices.
- `ic.index.column.id` (ColumnID) fornece a ordem das colunas dentro do índice.
- `ic.is.included.column` (IsIncludedColumn) indica se a coluna é incluída no índice.

Tabela `sys.columns` (c):

- Contém informações sobre as colunas de uma tabela.
- `c.name` (ColumnName) fornece o nome da coluna.

O índice `idx.dados.composite` é criado na tabela `dbo.dados` para otimizar consultas que envolvem as colunas `id.localre`, `id.sexo`, `id.grupoetario`, `id.condicaootrab`, `id.nivelescolaridade` e `id.ano`. A consulta de verificação lista todos os índices da tabela `dbo.dados`, mostrando detalhes como o nome do índice, o tipo, as colunas que fazem parte do índice e a sua ordem.

7.2 Function

A função `dbo.fn.calcular.estatisticas` é uma função definida pelo usuário que aceita vários parâmetros de entrada e retorna uma tabela com estatísticas calculadas com base nos dados fornecidos.

```
Function_fn_calcula...BD_censos (sa (70)) -> X
CREATE FUNCTION dbo.fn.calcular_estatisticas (
    @grupo_etario VARCHAR(255),
    @condicao_trabalho VARCHAR(255),
    @nivel_escolaridade VARCHAR(255),
    @localidade VARCHAR(255),
    @ano INT,
    @sexo VARCHAR(255)
)
RETURNS TABLE
AS
RETURN
(
    SELECT
        lr.descricao AS localidade,
        s.descricao AS sexo,
        ge.descricao AS grupo_etario,
        ct.descricao AS condicao_trabalho,
        ne.descricao AS nivel_escolaridade,
        a.ano,
        COUNT(d.id_dados) AS total_registros,
        AVG(d.valor) AS media_valores,
        SUM(d.valor) AS soma_valores
    FROM
        dbo.dados d
    JOIN
        dbo.localresidencia lr ON d.id_localre = lr.id_localre
    JOIN
        dbo.sexo s ON d.id_sexo = s.id_sexo
    JOIN
        dbo.grupoetario ge ON d.id_grupoetario = ge.id_grupoetario
    JOIN
        dbo.condicaotrab ct ON d.id_condicaotrab = ct.id_condicaotrab
    JOIN
        dbo.nivelescolaridade ne ON d.id_nivelescolaridade = ne.id_nivelescolaridade
    JOIN
        dbo.ano a ON d.id_ano = a.id_ano
    WHERE
        ge.descricao = @grupo_etario AND
        ct.descricao = @condicao_trabalho AND
        ne.descricao = @nivel_escolaridade AND
        lr.descricao = @localidade AND
        a.ano = @ano AND
        s.descricao = @sexo
    GROUP BY
        lr.descricao, s.descricao, ge.descricao, ct.descricao, ne.descricao, a.ano
    HAVING
        COUNT(d.id_dados) > 0
);
```

FIGURA 7.4: Criação da Function

A cláusula `SELECT` escolhe as colunas que serão retornadas:

- localidade: descrição da localidade;
- sexo: descrição do sexo;
- grupo.etario: descrição do grupo etário;
- condicao.trabalho: descrição da condição de trabalho;
- nivel.escolaridade: descrição do nível de escolaridade;
- ano: ano;
- total.registros: conta o número total de registros (`COUNT(d.id.dados)`);
- media.valores: calcula a média dos valores (`AVG(d.valor)`);
- soma.valores: calcula a soma dos valores (`SUM(d.valor)`).

A cláusula GROUP BY agrupa os dados pelas descrições e pelo ano, para calcular as estatísticas para cada grupo.

A cláusula HAVING filtra os grupos que têm pelo menos um registro (COUNT(d.id.dados) > 0).

```
INSERT INTO dbo.dados (id_localre, id_sexo, id_grupoetario, id_condicaotrab, id_nivelescolaridade, id_ano, valor)
VALUES
    ('1', '1', '01', '1', '3', 'S7A2011', 10),
    ('1', '1', '01', '1', '3', 'S7A2011', 20),
    ('1', '1', '01', '1', '3', 'S7A2011', 30), -- aumentar registros, para que a media seja diferente da soma
    ('1', '1', '01', '1', '3', 'S7A2021', 15),
    ('1', '1', '01', '1', '3', 'S7A2021', 25);

SELECT *
FROM dbo.fn_calcular_estatisticas('15 - 19 anos', 'População ativa', 'Ensino secundário', 'Continente', 2011, 'H');
```

localidade	sexo	grupo_etario	condicao_trabalho	nivel_escolaridade	ano	total_registros	media_valores	soma_valores
Continente	H	15 - 19 anos	População ativa	Ensino secundário	2011	4	2177	8708

FIGURA 7.5: Resultado da Function

Cinco novos registros são inseridos na tabela dbo.dados para o ano 'S7A2011', aumentando os registros para que a média dos valores seja diferente da soma.

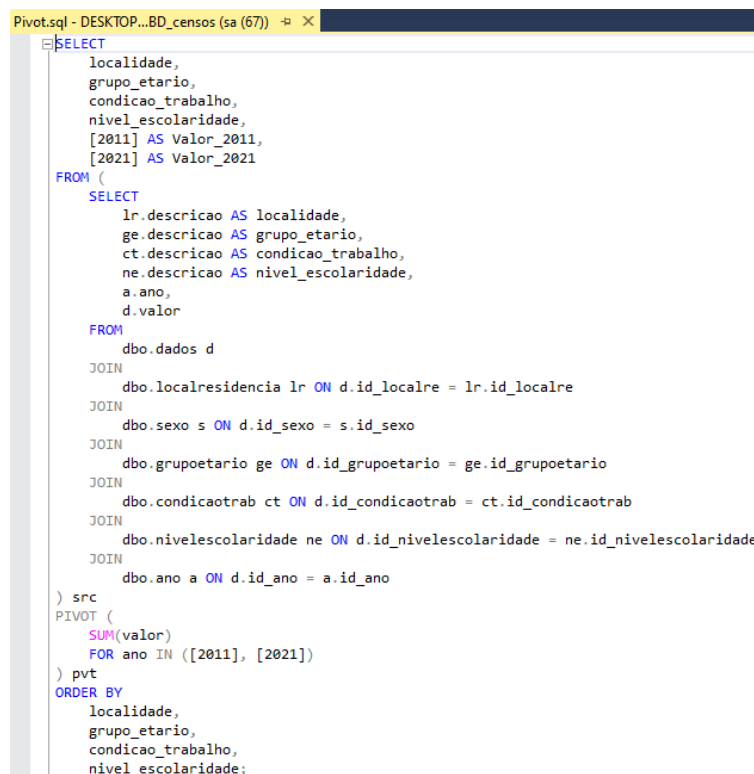
A função retorna uma linha com os dados agregados, mostrando a localidade, sexo, grupo etário, condição de trabalho, nível de escolaridade, ano, total de registros, média dos valores e soma dos valores para os parâmetros fornecidos.

A função dbo.fn.calcular.estatisticas é utilizada para calcular e retornar estatísticas (número total de registros, média e soma dos valores) para combinações específicas de grupo etário, condição de trabalho, nível de escolaridade, localidade, ano e sexo. Faz isto combinando várias tabelas de descrição e filtrando os resultados com base nos parâmetros de entrada. Os resultados são agrupados e filtrados para garantir que apenas grupos com pelo menos um registro sejam retornados.

7.3 PIVOT

O código SQL faz uma consulta numa base de dados para obter informações de diferentes tabelas, combinando-as com JOINS e, em seguida, usa uma cláusula PIVOT para reorganizar os dados.

O objetivo do código é gerar um relatório que mostre a soma dos valores (provavelmente populacionais ou estatísticos) para diferentes localidades, grupos etários, condições de trabalho e níveis de escolaridade, comparando os anos 2011 e 2021. A estrutura de PIVOT permite que os valores dos anos 2011 e 2021 sejam apresentados lado a lado em colunas separadas, facilitando a comparação.



```
Pivot.sql - DESKTOP...BD_censos (sa (67)) X
SELECT
    localidade,
    grupo_etario,
    condicao_trabalho,
    nivel_escolaridade,
    [2011] AS Valor_2011,
    [2021] AS Valor_2021
FROM (
    SELECT
        lr.descricao AS localidade,
        ge.descricao AS grupo_etario,
        ct.descricao AS condicao_trabalho,
        ne.descricao AS nivel_escolaridade,
        a.ano,
        d.valor
    FROM
        dbo.dados d
    JOIN
        dbo.localresidencia lr ON d.id_localre = lr.id_localre
    JOIN
        dbo.sexo s ON d.id_sexo = s.id_sexo
    JOIN
        dbo.grupoetario ge ON d.id_grupoetario = ge.id_grupoetario
    JOIN
        dbo.condicaotrab ct ON d.id_condicaotrab = ct.id_condicaotrab
    JOIN
        dbo.nivelescolaridade ne ON d.id_nivelescolaridade = ne.id_nivelescolaridade
    JOIN
        dbo.ano a ON d.id_ano = a.id_ano
) src
PIVOT (
    SUM(valor)
    FOR ano IN ([2011], [2021])
) pvt
ORDER BY
    localidade,
    grupo_etario,
    condicao_trabalho,
    nivel_escolaridade;
```

FIGURA 7.6: Criação de Pivot

Podemos ver que são selecionadas as colunas localidade, grupo.etário, condição.trabalho, nível.escolaridade, além dos valores de 2011 e 2021, que são renomeados para Valor.2011 e Valor.2021

A subconsulta existente combina várias tabelas (localresidencia, sexo, grupoetario, condicaotrab, nivelescolaridade, ano) com a tabela principal dados usando JOINS. Seleciona descrições de diferentes categorias e valores, juntamente com o ano.

A cláusula PIVOT transforma as linhas em colunas. Ela calcula a soma dos valores (SUM(valor)) para os anos 2011 e 2021, convertendo esses anos em colunas separadas [2011] e [2021].

Finalmente, os resultados são ordenados pelas colunas localidade, grupo.etario, condicao.trabalho e nivel.escolaridade.

Pivot.sql - [ReadO...BD_censos (sa (67))] SQLQuery1.sql - DE...BD_censos (sa (55))

```

SELECT
    localidade,
    grupo_etario,
    condicao_trabalho,
    nivel_escolaridade,
    [2011] AS Valor_2011,
    [2021] AS Valor_2021
FROM (
    SELECT
        lr.descricao AS localidade,
        ge.descricao AS grupo_etario,
        ct.descricao AS condicao_trabalho,
        ne.descricao AS nivel_escolaridade,
        a.ano,
        d.valor
    FROM
        dbo.dados d
    JOIN
        dbo.localresidencia lr ON d.id_localre = lr.id_localre
    JOIN
        dbo.sexo s ON d.id_sexo = s.id_sexo
    JOIN
        dbo.grupoetario ge ON d.id_grupoetario = ge.id_grupoetario
    JOIN
        dbo.condicaotrab ct ON d.id_condicaotrab = ct.id_condicaotrab

```

100 %

Results Messages

	localidade	grupo_etario	condicao_trabalho	nivel_escolaridade	Valor_2011	Valor_2021
1	Alentejo	15 - 19 anos	Desempregados	1.º ciclo	176	36
2	Alentejo	15 - 19 anos	Desempregados	2.º ciclo	744	144
3	Alentejo	15 - 19 anos	Desempregados	3.º ciclo	2094	422
4	Alentejo	15 - 19 anos	Desempregados	Bacharelato	0	0
5	Alentejo	15 - 19 anos	Desempregados	Curso técnico superior profissional	0	2
6	Alentejo	15 - 19 anos	Desempregados	Doutoramento	0	0
7	Alentejo	15 - 19 anos	Desempregados	Ensino básico	3014	602
8	Alentejo	15 - 19 anos	Desempregados	Ensino pós-secundário	60	0
9	Alentejo	15 - 19 anos	Desempregados	Ensino secundário	796	520
10	Alentejo	15 - 19 anos	Desempregados	Ensino superior	0	2
11	Alentejo	15 - 19 anos	Desempregados	Licenciatura	0	0
12	Alentejo	15 - 19 anos	Desempregados	Mestrado	0	0
13	Alentejo	15 - 19 anos	Desempregados	Nenhum	84	30
14	Alentejo	15 - 19 anos	Desempregados	Total	3954	1154
15	Alentejo	15 - 19 anos	Domésticos	1.º ciclo	74	28
16	Alentejo	15 - 19 anos	Domésticos	2.º ciclo	144	76
17	Alentejo	15 - 19 anos	Domésticos	3.º ciclo	176	88
18	Alentejo	15 - 19 anos	Domésticos	Bacharelato	0	0
19	Alentejo	15 - 19 anos	Domésticos	Curso técnico superior profissional	0	0

FIGURA 7.7: Resultado do Pivot

7.4 Consultas Simples

A consulta SQL faz o seguinte:

- Seleciona todos os registos e todas as colunas da VIEW vw.dados.simples;
- Filtra os registos para incluir apenas aqueles onde o sexo é masculino ('M');
- Ordena os registos filtrados pelo valor da coluna valor em ordem decrescente.

```
Instruções.sql - DE...ABD_censos (sa (83))
-- Filtrar Registros por Sexo e Ordenar por Valor
SELECT *
FROM vw_dados_simples
WHERE sexo = 'M'
ORDER BY valor DESC;
```

FIGURA 7.8: Código de criação da consulta

	id_dados	localidade	sexo	grupo_etario	condicao_trabalho	nivel_escolaridade	ano	valor
1	421400	Portugal	M	Total	Total	Total	2021	4773902
2	209720	Portugal	M	Total	Total	Total	2012	4747248
3	215600	Continente	M	Total	Total	Total	2021	4552324
4	3920	Continente	M	Total	Total	Total	2012	4523544
5	209708	Portugal	M	Total	Total	Ensino básico	2012	2607740
6	3908	Continente	M	Total	Total	Ensino básico	2012	2475893
7	209594	Portugal	M	Total	População ativa	Total	2012	2419793
8	421316	Portugal	M	Total	População inativa	Total	2021	2391392
9	421274	Portugal	M	Total	População ativa	Total	2021	2382510
10	209636	Portugal	M	Total	População inativa	Total	2012	2327455
11	3794	Continente	M	Total	População ativa	Total	2012	2308328
12	215516	Continente	M	Total	População inativa	Total	2021	2280270
13	215474	Continente	M	Total	População ativa	Total	2021	2272054
14	421388	Portugal	M	Total	Total	Ensino básico	2021	2238315
15	3836	Continente	M	Total	População inativa	Total	2012	2215216
16	421288	Portugal	M	Total	Empregados	Total	2021	2169935
17	215588	Continente	M	Total	Total	Ensino básico	2021	2125779
18	209608	Portugal	M	Total	Empregados	Total	2012	2085213
19	215488	Continente	M	Total	Empregados	Total	2021	2069996
20	3808	Continente	M	Total	Empregados	Total	2012	1986962
21	221480	Norte	M	Total	Total	Total	2021	1664188
22	9800	Norte	M	Total	Total	Total	2012	1651042
23	421304	Portugal	M	Total	População inativa	Ensino básico	2021	1472149
24	209624	Portugal	M	Total	População inativa	Ensino básico	2012	1436597
25	215504	Continente	M	Total	População inativa	Ensino básico	2021	1401153
26	3824	Continente	M	Total	População inativa	Ensino básico	2012	1363162
27	344960	Área Met...	M	Total	Total	Total	2021	1319472
28	339080	Área Met...	M	Total	Total	Total	2021	1319472
29	421358	Portugal	M	Total	Reformados, ap...	Total	2021	1317122
30	209678	Portugal	M	Total	Reformados, ap...	Total	2012	1298984
31	209709	Portugal	M	Total	Total	1.º ciclo	2012	1278567
32	133280	Área Met...	M	Total	Total	Total	2012	1272966
33	127400	Área Met...	M	Total	Total	Total	2012	1272966
34	215558	Continente	M	Total	Reformados, ap...	Total	2021	1269011
35	3878	Continente	M	Total	Reformados, ap...	Total	2012	1251194
36	3909	Continente	M	Total	Total	1.º ciclo	2012	1214894
37	209582	Portugal	M	Total	População ativa	Ensino básico	2012	1171143
38	3782	Continente	M	Total	População ativa	Ensino básico	2012	1112731
39	421389	Portugal	M	Total	Total	1.º ciclo	2021	1102958
40	421392	Portugal	M	Total	Total	Ensino secundário	2021	1075484
41	421394	Portugal	M	Total	Total	Ensino superior	2021	1070106
42	74480	Centro	M	Total	Total	Total	2012	1060618

FIGURA 7.9: Resultado da Consulta

Esta consulta é útil para obter uma lista de todos os registos masculinos na VIEW vw.dados.simples e classificá-los do maior para o menor valor.

O objetivo desta consulta SQL é obter uma lista de registos da tabela dbo.dados e das suas tabelas associadas (localresidencia, sexo, grupoetario, condicaotrab, nivelescolaridade, ano) onde o valor seja maior que 1.000.000. Os resultados são apresentados com as descrições completas de localidade, sexo, grupo etário, condição de trabalho, nível de escolaridade e ano, e ordenados pelo valor em ordem crescente.

```

Instruções.sql - DE...ABD_censos (sa (83))
-- Instrução SQL - Valor > 1 000 000
SELECT
  d.id_dados,
  lr.descricao AS localidade,
  s.descricao AS sexo,
  ge.descricao AS grupo_etario,
  ct.descricao AS condicao_trabalho,
  ne.descricao AS nivel_escolaridade,
  a.ano,
  d.valor
FROM
  dbo.dados d
JOIN
  dbo.localresidencia lr ON d.id_localre = lr.id_localre
JOIN
  dbo.sexo s ON d.id_sexo = s.id_sexo
JOIN
  dbo.grupoetario ge ON d.id_grupoetario = ge.id_grupoetario
JOIN
  dbo.condicaotrab ct ON d.id_condicaotrab = ct.id_condicaotrab
JOIN
  dbo.nivelescolaridade ne ON d.id_nivelescolaridade = ne.id_nivelescolaridade
JOIN
  dbo.ano a ON d.id_ano = a.id_ano
WHERE
  d.valor > 1000000
ORDER BY
  valor;

```

FIGURA 7.10: Código de criação da consulta

Essa consulta pode ser utilizada em cenários onde é necessário:

- Analisar registos com valores excepcionalmente altos (acima de 1.000.000);
- Obter uma visão detalhada das características associadas a esses registos;
- Facilitar a revisão e análise de dados estatísticos ou financeiros que excedem um determinado limite, permitindo identificar tendências, padrões ou anomalias nos dados.

id_dados	localidade	sexo	grupo_etario	condicao_trabalho	nivel_escolaridade	ano	valor	
1	5742	Continente	HM	Total	População ativa	2012	103558	
2	5023	Nome	Homem	Total	Erano básico	2012	1003010	
3	1918	Continente	Homem	Total	Reformados, aposentados ou na reserva	2012	1005951	
4	341026	Área Metropolitana de Lisboa	HM	Total	Erano básico	2021	1013885	
5	340903	Área Metropolitana de Lisboa	HM	Total	Erano básico	2021	1013885	
6	215592	Continente	M	Total	Erano secundário	2021	1027789	
7	215594	Continente	M	Total	Erano superior	2021	1026591	
8	288108	Centro	HM	Total	Erano básico	2021	1030305	
9	211545	Portugal	HM	Total	População ativa	2012	1037989	
10	288108	Centro	M	Total	Total	2021	1038176	
11	207718	Portugal	Homem	Total	Reformados, aposentados ou na reserva	2012	1040110	
12	211545	Portugal	HM	Total	População ativa	2012	1043395	
13	5185	Continente	HM	Total	População ativa	21.º ciclo	2012	1046476
14	215589	Continente	M	Total	Total	1.º ciclo	2021	1048062
15	11622	Nome	HM	Total	População ativa	Erano básico	2012	1050579
16	30714	Centro	HM	Total	População ativa	Total	2012	1056225
17	74480	Centro	M	Total	Total	2012	1060618	
18	217178	Continente	HM	75 ou mais	Reformados, aposentados ou na reserva	Total	2021	1062096
19	421390	Portugal	M	Total	Total	Erano superior	2021	1070106
20	421392	Portugal	M	Total	Total	Erano secundário	2021	1074804
21	340995	Área Metropolitana de Lisboa	HM	Total	População inativa	Total	2021	1081229
22	340926	Área Metropolitana de Lisboa	HM	Total	População inativa	Total	2021	1081229
23	5870	Continente	HM	Total	Total	2.º ciclo	2012	1083685
24	1964	Continente	Homem	Total	População inativa	Erano básico	2012	1087919
25	211545	Portugal	HM	Total	População ativa	3.º ciclo	2012	1093396
26	421718	Portugal	HM	75 ou mais	Reformados, aposentados ou na reserva	Total	2021	1097107
27	421389	Portugal	M	Total	Total	1.º ciclo	2021	1102969
28	1949	Continente	Homem	Total	Total	1.º ciclo	2012	1109433
29	125440	Área Metropolitana de Lisboa	Homem	Total	Total	Total	2012	1111029
30	101320	Área Metropolitana de Lisboa	Homem	Total	Total	Total	2012	1111029
31	3762	Continente	M	Total	População ativa	Erano básico	2012	1112711
32	217128	Continente	HM	75 ou mais	População inativa	Total	2021	1124593
33	5027	Continente	HM	Total	Reformados, aposentados ou na reserva	1.º ciclo	2012	1126326
34	207864	Portugal	Homem	Total	População inativa	Erano básico	2012	1136480
35	217420	Continente	HM	75 ou mais	Total	Total	2021	1140052
36	211670	Portugal	HM	Total	Total	2.º ciclo	2012	1153257
37	217557	Continente	HM	Total	Total	Leontatura	2021	1160374
38	421390	Portugal	HM	75 ou mais	População inativa	Total	2021	1165624
39	207749	Portugal	Homem	Total	Total	1.º ciclo	2012	1167827
40	211627	Portugal	HM	Total	Reformados, aposentados ou na reserva	1.º ciclo	2012	1168905
41	205552	Portugal	M	Total	População ativa	Erano básico	2012	1171143
42	207930	Portugal	HM	75 ou mais	Total	Total	2021	1176411

FIGURA 7.11: Resultado da Consulta

Capítulo 8

Otimização

Para analisar o desempenho da base de dados foi usada uma query relativamente pesada em termos de execução que passou num tracer. Um tracer é mecanismo que captura e registra informações sobre eventos e atividades que ocorrem na base de dados. Essas informações podem incluir consultas executadas, transações concluídas, erros encontrados para monitorizar e analisar o desempenho do sistema.

8.1 Configuração do Tracer

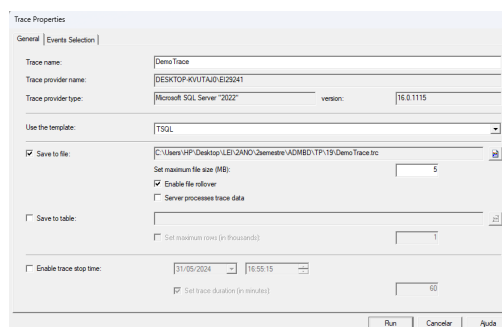


FIGURA 8.1: Na página Trace Properties na aba General definir o nome do tracer, escolher o Template TSQL, ativar a opção Save to File e escolher o caminho onde o ficheiro vai ser guardado

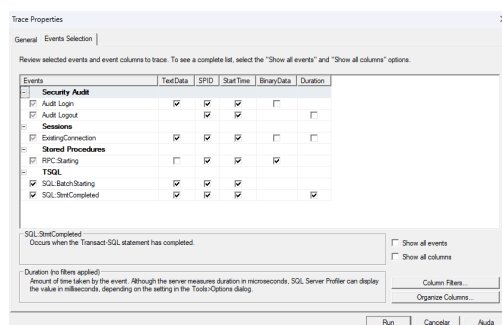


FIGURA 8.2: Na página Trace Properties na aba Events Selection seleccionar o evento SQL:StmtCompleted e a coluna Duration apenas para este evento

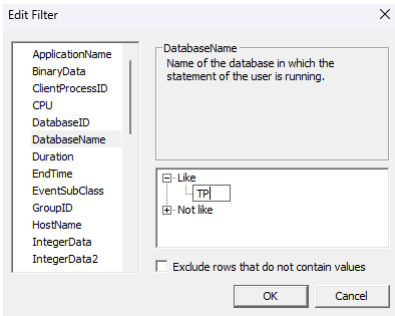


FIGURA 8.3: Duplo clique na coluna DatabaseName expandir a propriedade Like e escrever o nome da base de dados

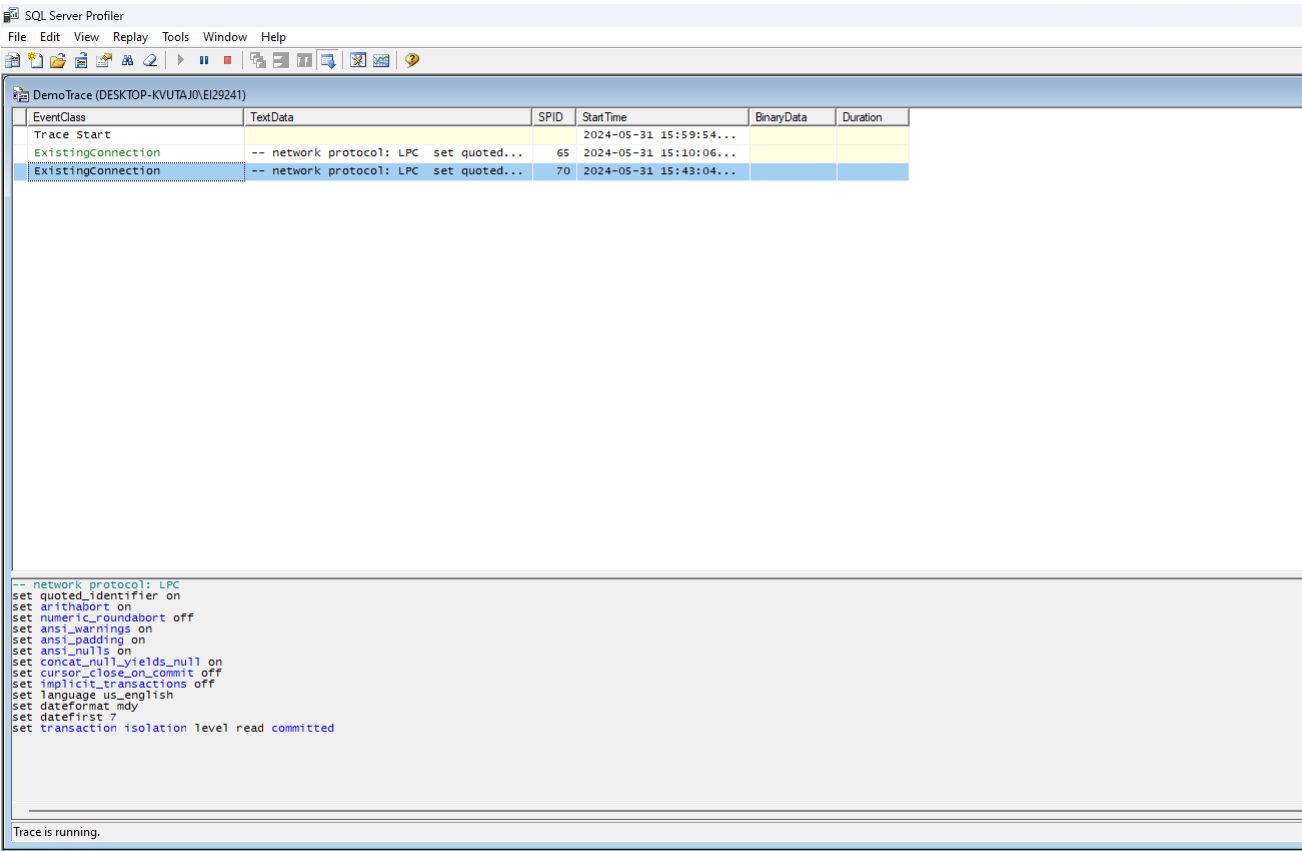


FIGURA 8.4: Verificar que o Tracer está a correr

```
USE TP;
GO

SELECT SUM(d.valor) AS total, a.ano, ct.descricao, ge.descricao, lr.descricao, ne.descricao, s.descricao
FROM dbo.dados d
JOIN dbo.ano a ON d.id_ano = a.id_ano
JOIN dbo.condicaoatrab ct ON d.id_condicaoatrab = ct.id_condicaoatrab
JOIN dbo.grupoetario ge ON d.id_grupoetario = ge.descricao
JOIN dbo.localresidencia lr ON d.id_localre = lr.id_localre
JOIN dbo.nivelescolaridade ne ON d.id_nivelescolaridade = ne.id_nivelescolaridade
JOIN dbo.sexo s ON d.id_sexo = s.id_sexo
WHERE d.id_localre = '118'
AND d.id_sexo = '2'
GROUP BY a.ano, ct.descricao, ge.descricao, lr.descricao, ne.descricao, s.descricao

GO 28
```

FIGURA 8.5: Query usada para testar o desempenho

DemoTrace (DESKTOP-KVUTAJ0\EI29241)

EventClass	TextData	SPID	StartTime	BinaryData	Duration
SQL:BatchStarting	SELECT SUM(d.valor), a.ano, ct.des...	70	2024-05-31 16:09:33...		
SQL:StmtCompleted	SELECT SUM(d.valor), a.ano, ct.des...	70	2024-05-31 16:09:33...		90
SQL:BatchStarting	SELECT SUM(d.valor), a.ano, ct.des...	70	2024-05-31 16:09:33...		
SQL:StmtCompleted	SELECT SUM(d.valor), a.ano, ct.des...	70	2024-05-31 16:09:33...		93
SQL:BatchStarting	SELECT SUM(d.valor), a.ano, ct.des...	70	2024-05-31 16:09:33...		
SQL:StmtCompleted	SELECT SUM(d.valor), a.ano, ct.des...	70	2024-05-31 16:09:33...		94
SQL:BatchStarting	SELECT SUM(d.valor), a.ano, ct.des...	70	2024-05-31 16:09:33...		
SQL:StmtCompleted	SELECT SUM(d.valor), a.ano, ct.des...	70	2024-05-31 16:09:33...		96
SQL:BatchStarting	SELECT SUM(d.valor), a.ano, ct.des...	70	2024-05-31 16:09:33...		
SQL:StmtCompleted	SELECT SUM(d.valor), a.ano, ct.des...	70	2024-05-31 16:09:33...		96
SQL:BatchStarting	SELECT SUM(d.valor), a.ano, ct.des...	70	2024-05-31 16:09:34...		
SQL:StmtCompleted	SELECT SUM(d.valor), a.ano, ct.des...	70	2024-05-31 16:09:34...		95
SQL:BatchStarting	SELECT SUM(d.valor), a.ano, ct.des...	70	2024-05-31 16:09:34...		
SQL:StmtCompleted	SELECT SUM(d.valor), a.ano, ct.des...	70	2024-05-31 16:09:34...		96
SQL:BatchStarting	SELECT SUM(d.valor), a.ano, ct.des...	70	2024-05-31 16:09:34...		
SQL:StmtCompleted	SELECT SUM(d.valor), a.ano, ct.des...	70	2024-05-31 16:09:34...		117
SQL:BatchStarting	SELECT SUM(d.valor), a.ano, ct.des...	70	2024-05-31 16:09:34...		
SQL:StmtCompleted	SELECT SUM(d.valor), a.ano, ct.des...	70	2024-05-31 16:09:34...		87
SQL:BatchStarting	SELECT SUM(d.valor), a.ano, ct.des...	70	2024-05-31 16:09:34...		
SQL:StmtCompleted	SELECT SUM(d.valor), a.ano, ct.des...	70	2024-05-31 16:09:34...		97
SQL:BatchStarting	SELECT SUM(d.valor), a.ano, ct.des...	70	2024-05-31 16:09:34...		
SQL:StmtCompleted	SELECT SUM(d.valor), a.ano, ct.des...	70	2024-05-31 16:09:34...		94
SQL:BatchStarting	SELECT SUM(d.valor), a.ano, ct.des...	70	2024-05-31 16:09:34...		
SQL:StmtCompleted	SELECT SUM(d.valor), a.ano, ct.des...	70	2024-05-31 16:09:34...		91

SELECT SUM(d.valor), a.ano, ct.descricao, ge.descricao, lr.descricao, ne.descricao, s.descricao
FROM dbo.dados d
JOIN dbo.ano a ON d.id_ano = a.id_ano
JOIN dbo.condicaoatrab ct ON d.id_condicaoatrab = ct.id_condicaoatrab
JOIN dbo.grupoetario ge ON d.id_grupoetario = ge.descricao
JOIN dbo.localresidencia lr ON d.id_localre = lr.id_localre
JOIN dbo.nivelescolaridade ne ON d.id_nivelescolaridade = ne.id_nivelescolaridade
JOIN dbo.sexo s ON d.id_sexo = s.id_sexo
WHERE d.id_localre = '118'
AND d.id_sexo = '2'
GROUP BY a.ano, ct.descricao, ge.descricao, lr.descricao, ne.descricao, s.descricao

Trace is running.

FIGURA 8.6: Verificar que a query passou no tracer

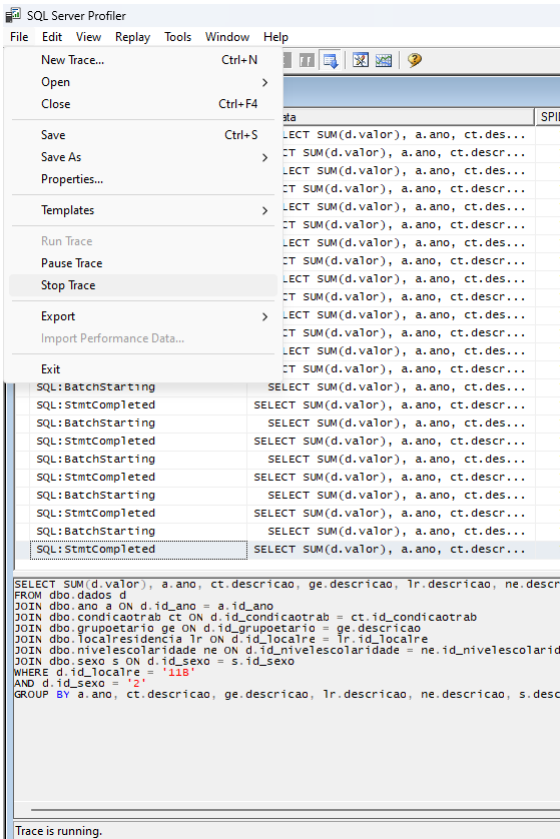


FIGURA 8.7: Parar o tracer

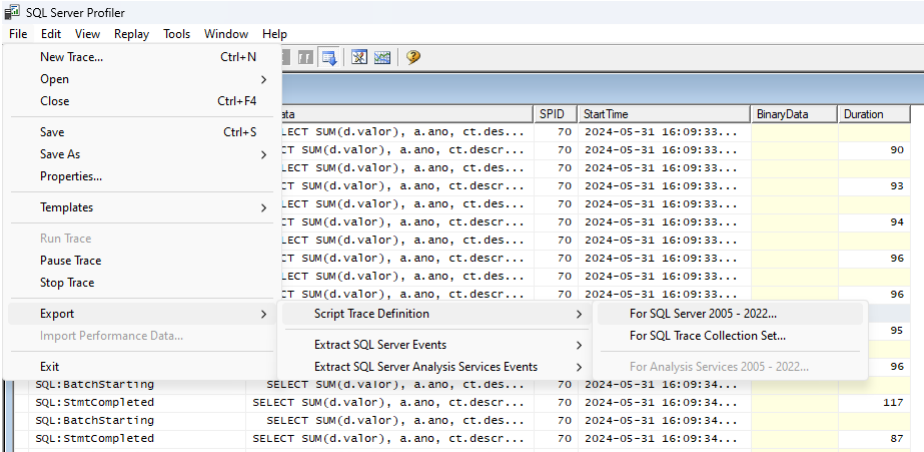


FIGURA 8.8: Exportar o tracer

8.2 Analisar desempenho da base de dados

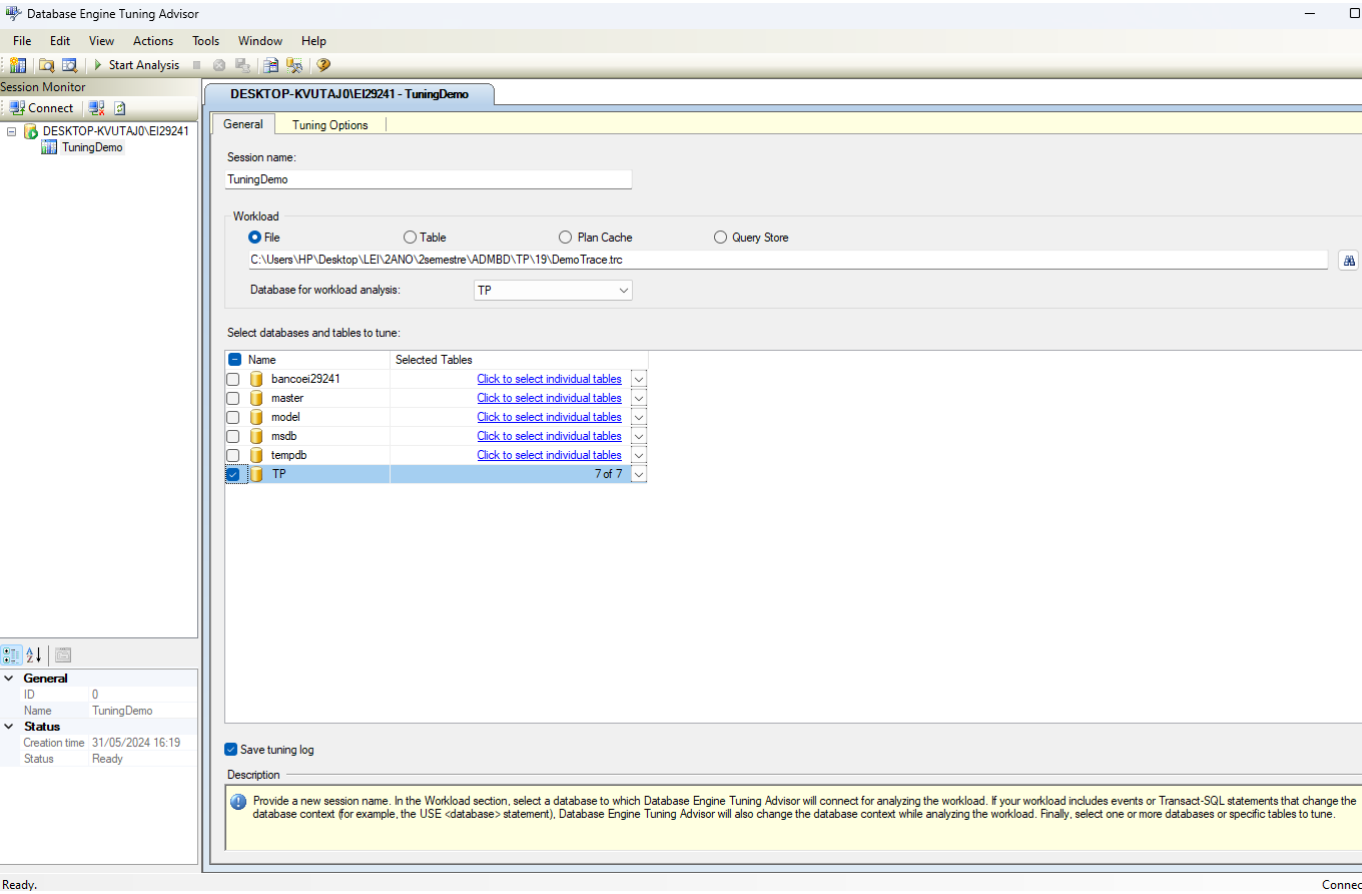


FIGURA 8.9: Depois de abrir o Database Engine Tuning Advisor e conectar à instancia na aba General definir o nome da sessão escolher o caminho onde foi guardado o tracer e escolher a base de dados

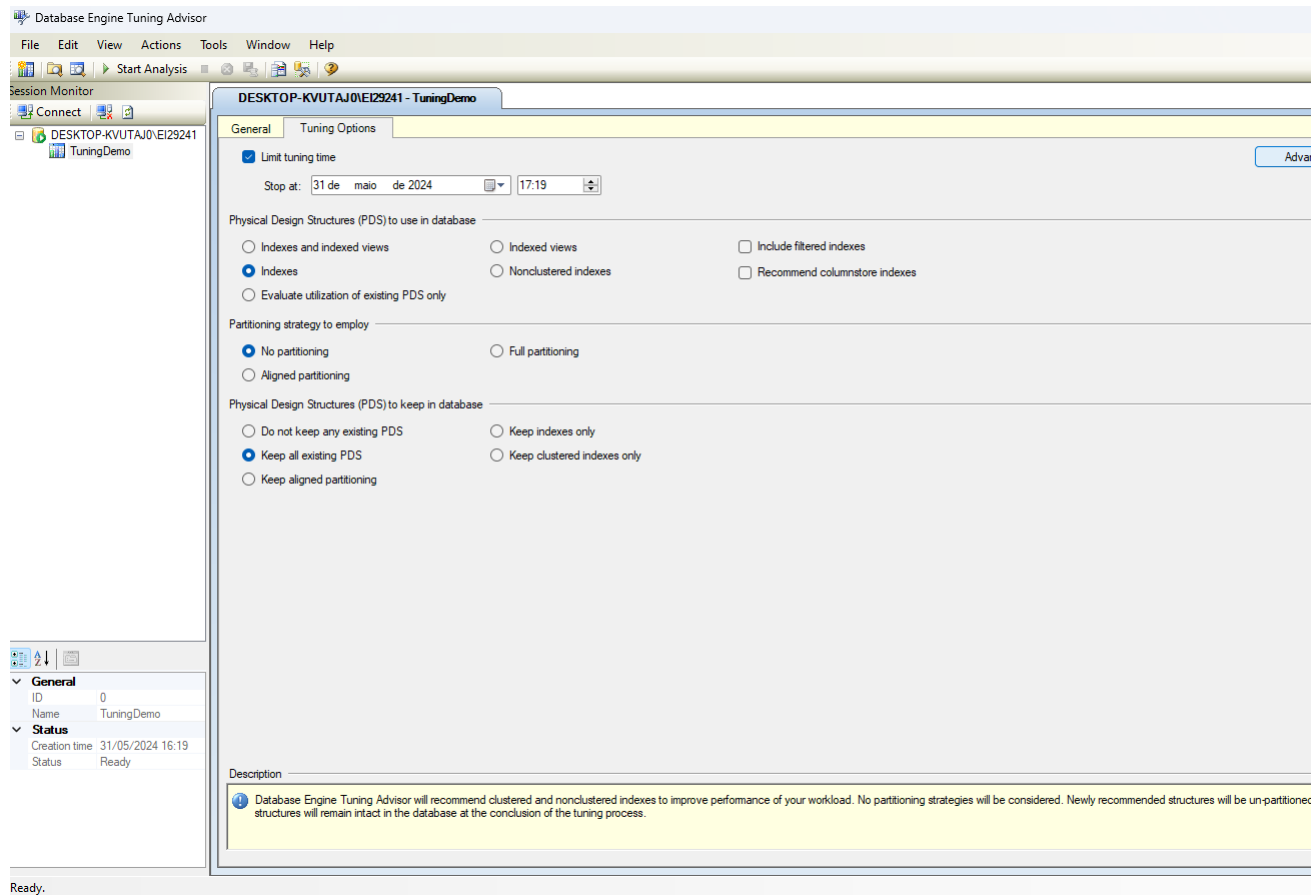


FIGURA 8.10: Na aba Tuning Options clicar em Advanced Options

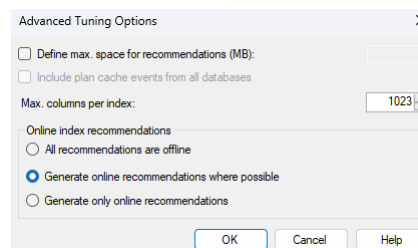


FIGURA 8.11: Na página Advanced Tuning Options selecionar Generate online recommendations where possible

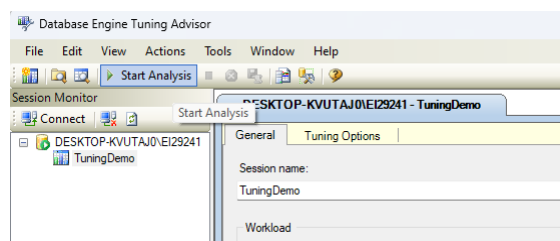


FIGURA 8.12: Começar a análise

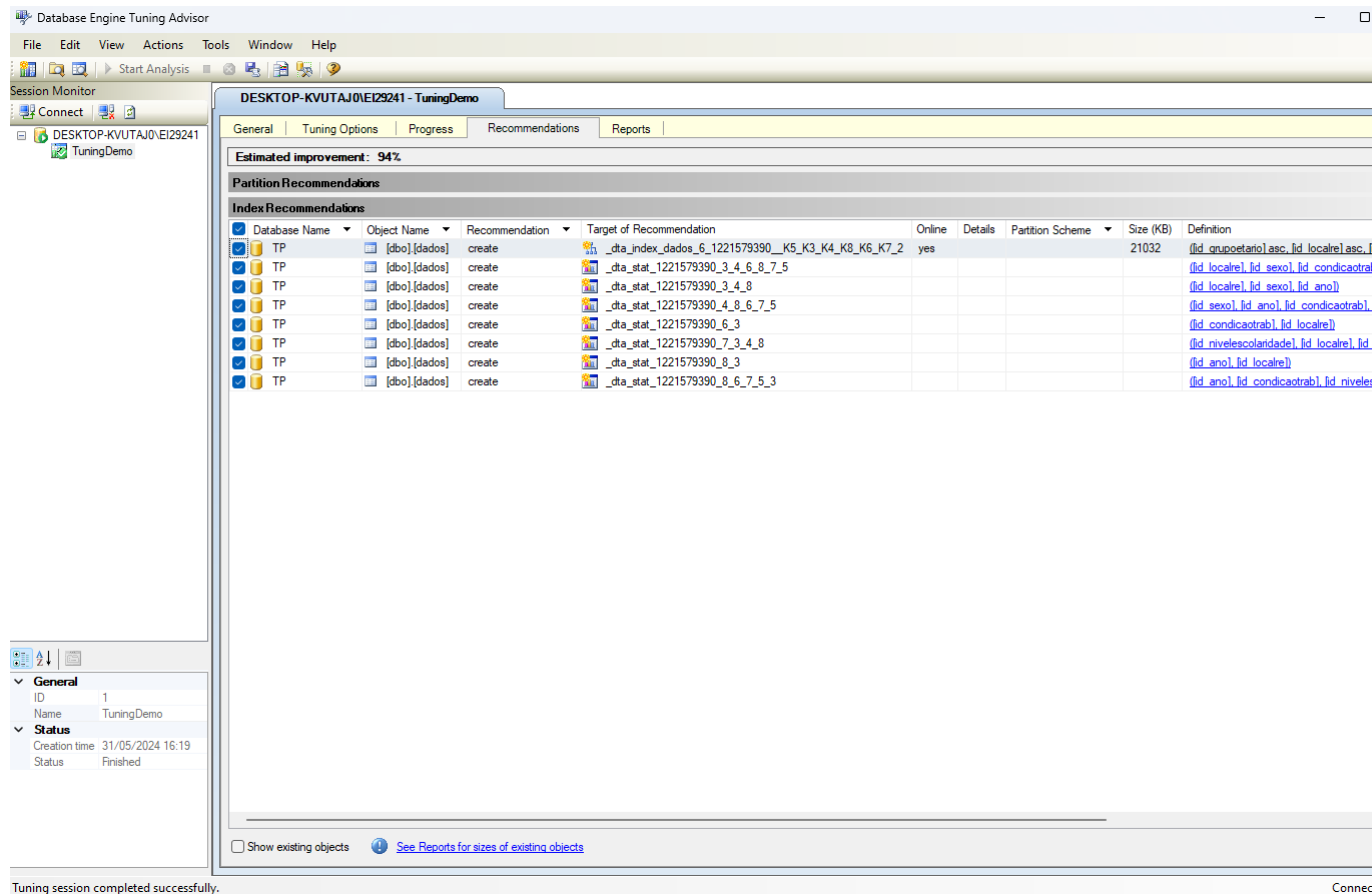


FIGURA 8.13: Aceder à aba Recommendations e analisar as recomendações

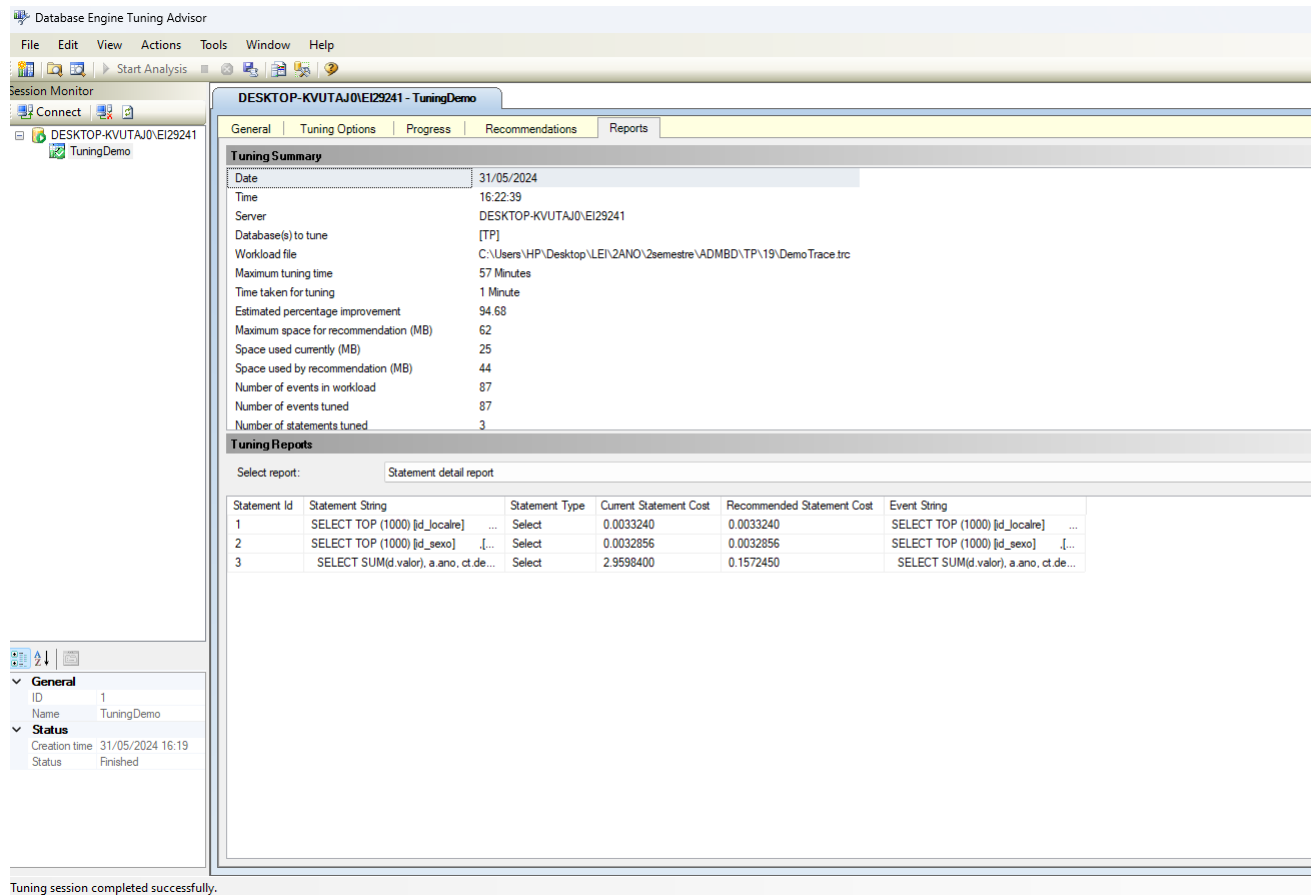


FIGURA 8.14: Na aba Reports analisar os valores Current Statement Cost e Recommended Statement Cost, existe uma grande diferença

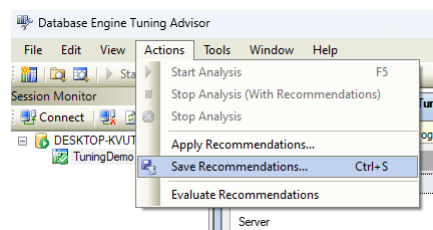


FIGURA 8.15: Guardar Recomendações

8.3 Melhorar desempenho da base de dados

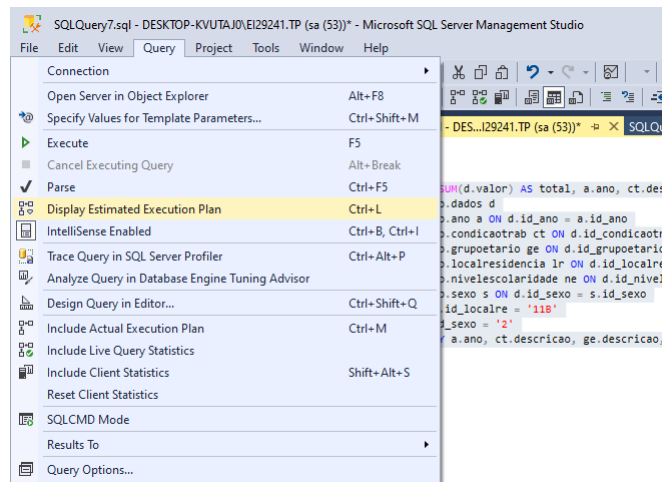


FIGURA 8.16: Analisar plano de execução da query que passou no tracer, exceto o GO 20

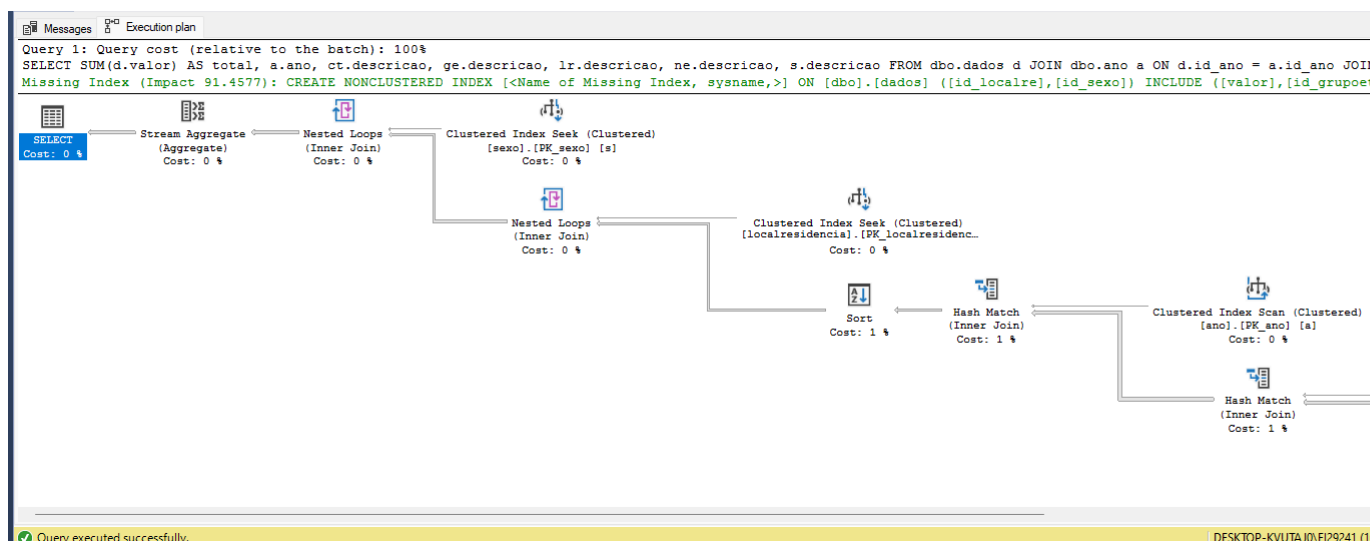


FIGURA 8.17: Verificar que há índices em falta

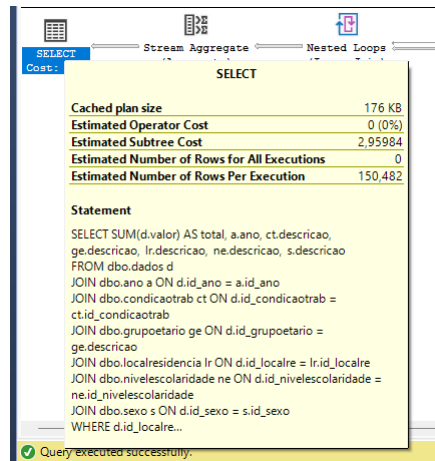


FIGURA 8.18: Observar Estimated Subtree Cost

```
use [TP]
go

SET ANSI_PADDING ON
go

CREATE NONCLUSTERED INDEX [_dta_index_dados_6_1221579390_K5_K3_K4_K8_K6_K7_2] ON [dbo].[dados]
(
    [id_grupoetario] ASC,
    [id_localre] ASC,
    [id_sexo] ASC,
    [id_ano] ASC,
    [id_condicaootrab] ASC,
    [id_nivelescolaridade] ASC
)
INCLUDE([valor]) WITH (SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = ON) ON [PRIMARY]
go

CREATE STATISTICS [_dta_stat_1221579390_8_3] ON [dbo].[dados]([id_ano], [id_localre])
WITH AUTO_DROP = OFF
go

CREATE STATISTICS [_dta_stat_1221579390_6_3] ON [dbo].[dados]([id_condicaootrab], [id_localre])
WITH AUTO_DROP = OFF
go

CREATE STATISTICS [_dta_stat_1221579390_7_3_4_8] ON [dbo].[dados]([id_localre], [id_sexo], [id_ano])
WITH AUTO_DROP = OFF
go

CREATE STATISTICS [_dta_stat_1221579390_7_3_4_8] ON [dbo].[dados]([id_nivelescolaridade], [id_localre], [id_sexo], [id_ano])
WITH AUTO_DROP = OFF
go

CREATE STATISTICS [_dta_stat_1221579390_8_6_7_5_3] ON [dbo].[dados]([id_ano], [id_condicaootrab], [id_nivelescolaridade], [id_grupoetario], [id_localre])
WITH AUTO_DROP = OFF
go

CREATE STATISTICS [_dta_stat_1221579390_4_8_6_7_5] ON [dbo].[dados]([id_sexo], [id_ano], [id_condicaootrab], [id_nivelescolaridade], [id_grupoetario])
WITH AUTO_DROP = OFF
go

CREATE STATISTICS [_dta_stat_1221579390_3_4_6_8_7_5] ON [dbo].[dados]([id_localre], [id_sexo], [id_condicaootrab], [id_ano], [id_nivelescolaridade], [id_grupoetario])
WITH AUTO_DROP = OFF
go
```

FIGURA 8.19: Executar o script para aplicar as recomendações

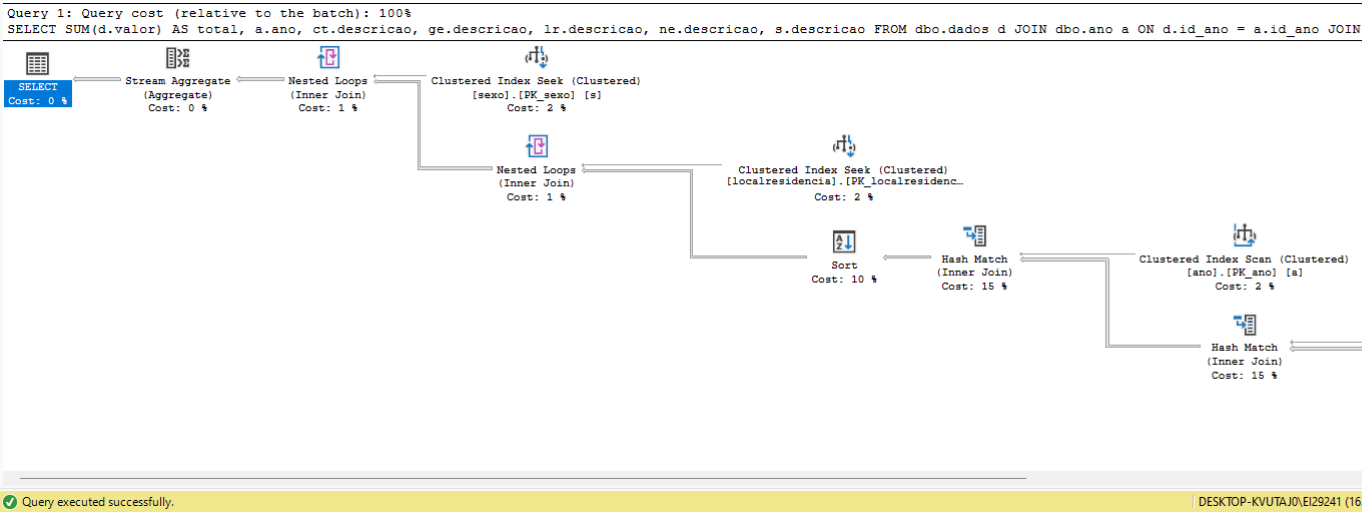


FIGURA 8.20: Observar que já não há falta de índices

Capítulo 9

Relatório gerado pelo Report Builder

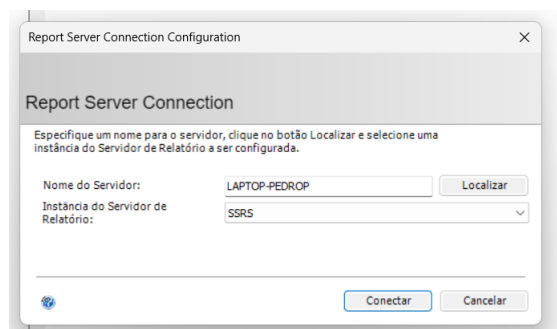


FIGURA 9.1: Conectar á instância no report server configuration manager



FIGURA 9.2: Informações relativas ao servidor atual

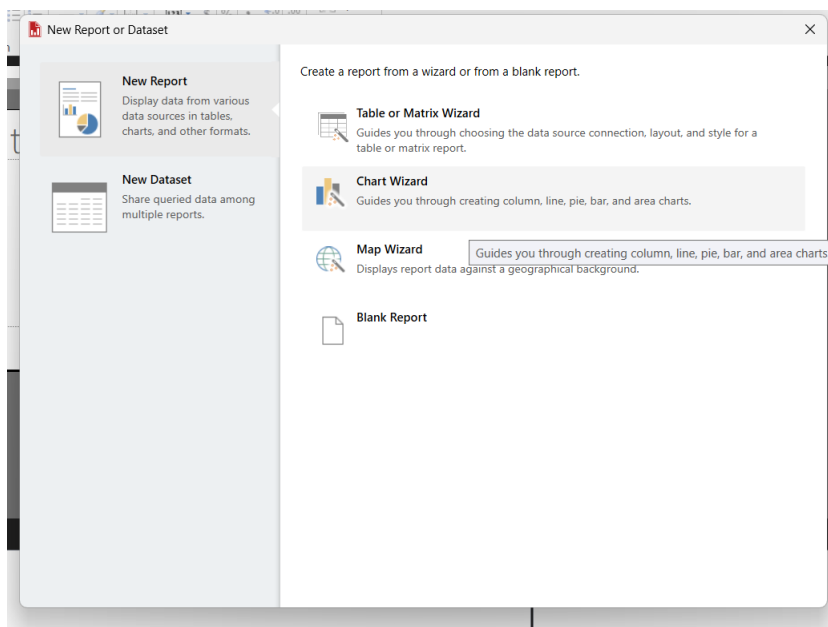


FIGURA 9.3: Com o report builder aberto criar um novo relatório em tabela

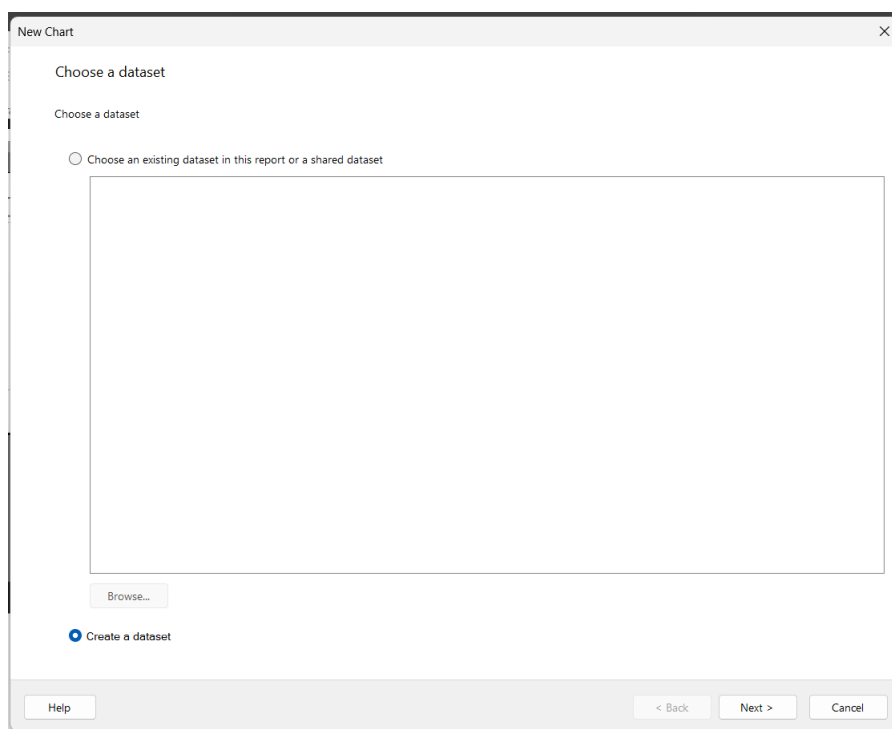


FIGURA 9.4: Criar um novo dataset

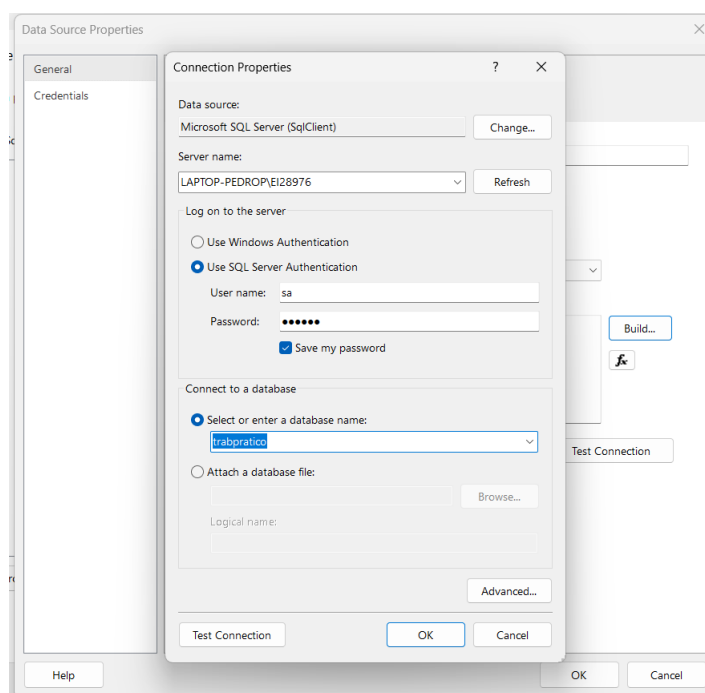


FIGURA 9.5: Conexão á base de dados

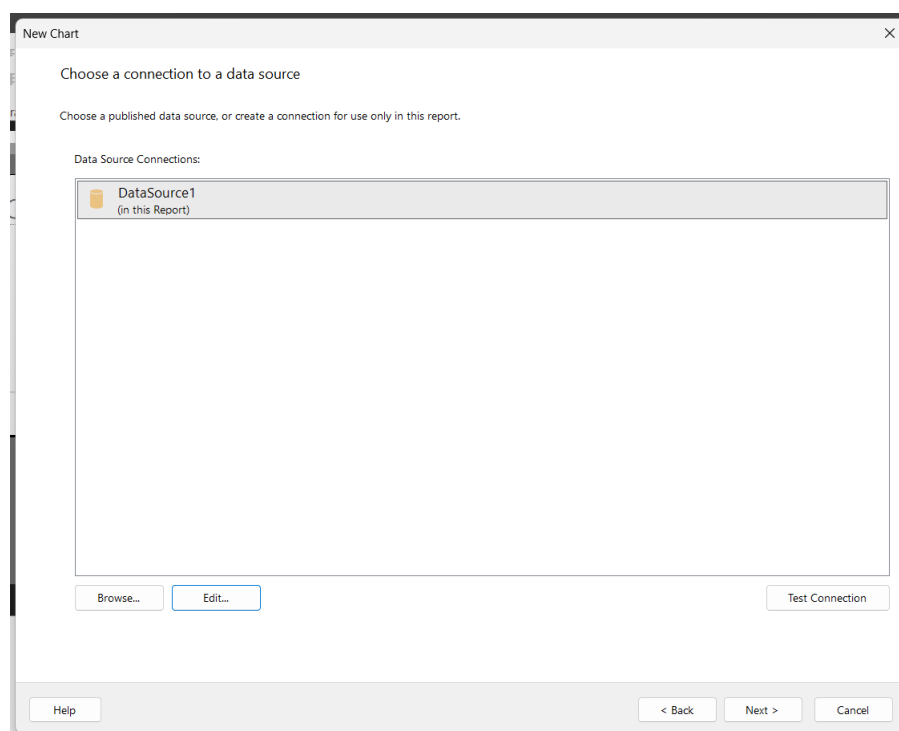


FIGURA 9.6: Selecionar dataset criado

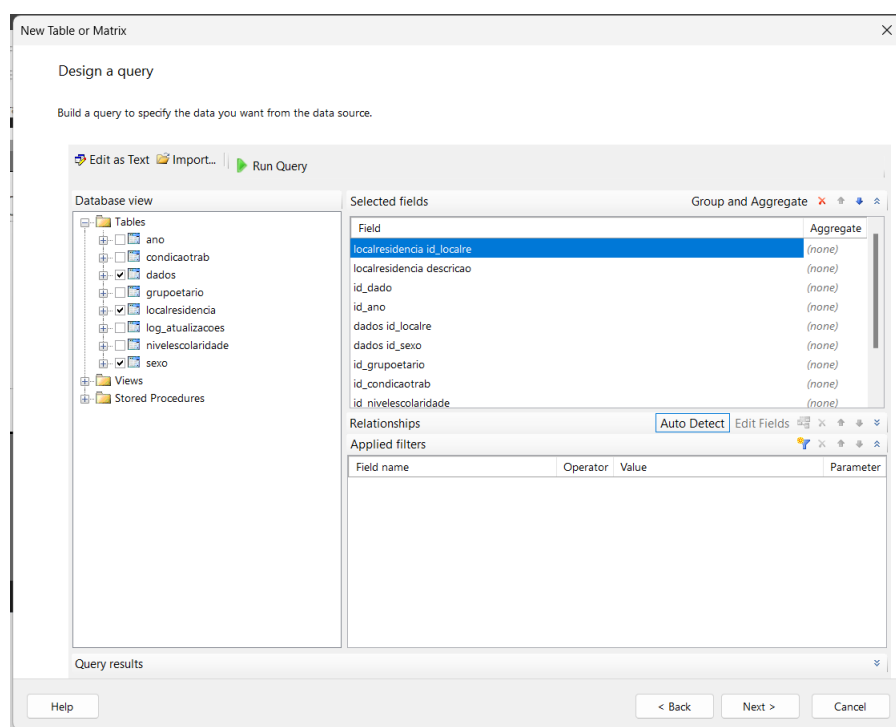


FIGURA 9.7: Seleção das tabelas que se vão utilizar

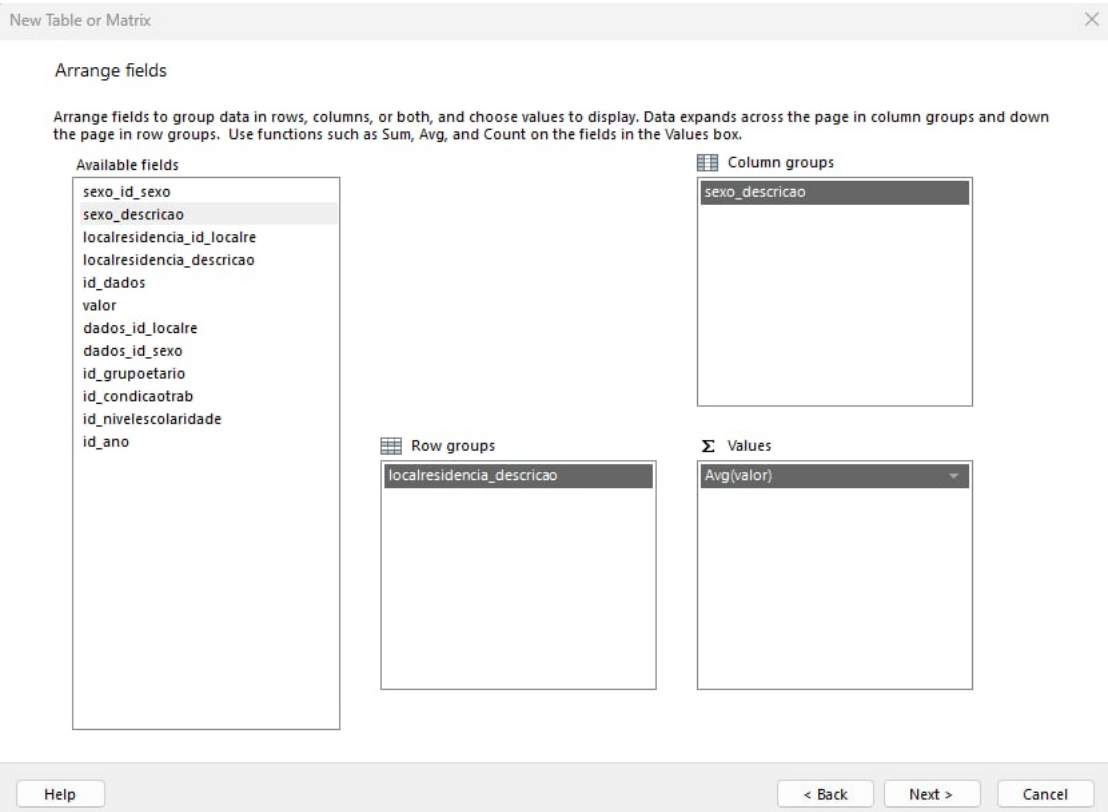


FIGURA 9.8: Com sexo_ddescricao nas colunas, nas linhas o localresidencia_ddescricao e a mediada dos valores obteremos a

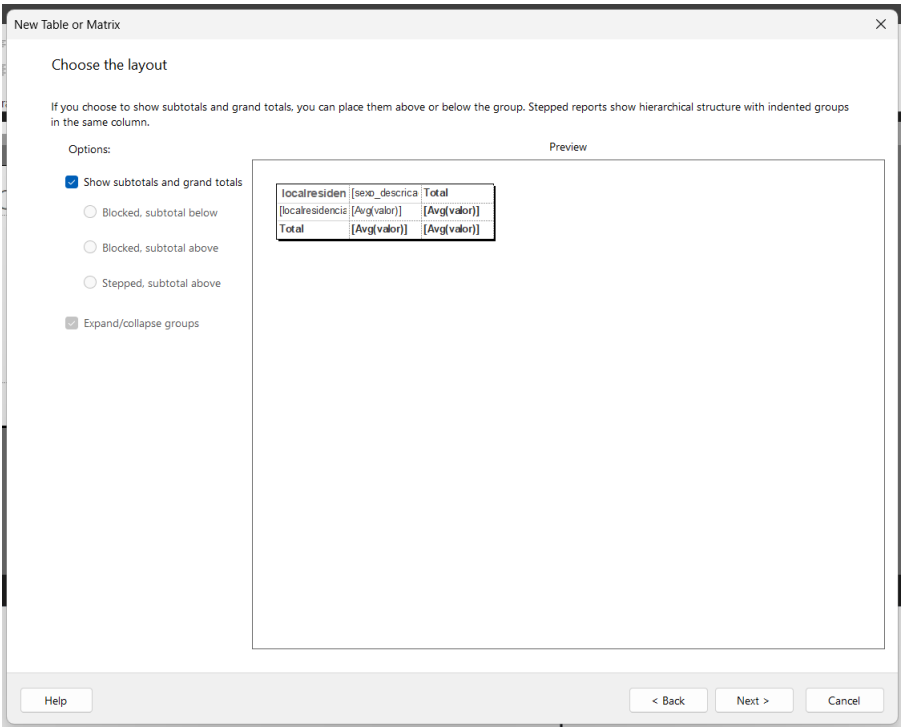


FIGURA 9.9: Visualização do formato da tabela

Localidade	Residência	H	M	Total
Alentejo	2530,0239795	5220,2372448	2690,2132653	3480,1581632
	9184	9796	0612	6531
Alentejo Central	548,05561224	1143,3719387	595,31632653	762,24795918
	4898	7551	0612	3674
Alentejo Litoral	350,02806122	692,11683673	342,08877551	461,41122448
	449	4694	0204	9796
Algarve	1562,8040816	3223,9010204	1661,0969387	2149,2673469
	3265	0816	7551	3878
Alto Alentejo	384,55255102	804,75459183	420,20204081	536,50306122
	0408	6735	6327	449
Alto Minho	775,69348855	1739,4540816	925,92448979	1110,2835980
	0595	3265	5918	9981
Alto Tâmega	322,26020408	565,62168458	348,27755102	409,39625144
	1633	7814	0408	1753
Área Metropolitana de Lisboa	9309,4484693	20028,463775	10719,015306	13352,309183
	8775	5102	1224	6735
Área Metropolitana do Porto	6030,8240546	12627,121428	6671,1775510	8466,2354052
	2185	5714	2041	1978
Ave	1460,7112244	3036,5112244	1575,8	2024,3408163
	898	898		2653
Baixo Alentejo	424,00102040	861,94438775	437,94336734	574,62959183
	8163	5102	6939	6735
Beira Baixa	297,02908163	623,31581632	326,28673469	415,54387755
	2653	6531	3878	102
Beiras e Serra da Estrela	785,51020408	1652,1122448	866,60204081	1101,4081632
	1633	9796	6327	6531
Cavado	1394,0397959	2925,0306122	1530,9908163	1950,0204081
	1837	449	2653	6327
Centro	7830,9244897	16476,249489	8645,325	10984,166326
	9592	7959		5306
Continente	33812,508163	71317,173469	37504,665306	47544,782312
	2653	3878	1224	9252
Douro	676,63775510	1420,0806122	743,44285714	946,72040816
	2041	449	2857	3265
Lezíria do Tejo	823,38673469	1718,0494897	894,66275510	1145,3663265
	3878	9592	2041	3061
Médio Tejo	815,22397959	1721,5239795	906,3	1147,6826530
	1837	9184		6122

02/06/2024 15:26:44

FIGURA 9.10: Visualização do relatório gerado

Norte	12502,238775	26253,684183	14334,884918	18256,489611
	5102	6735	4783	0057
Oeste	1240,2306122	2579,1581632	1338,9275510	1719,4387755
	449	6531	2041	102
Portugal	35374,640816	72432,581122	39356,837755	49054,686564
	3265	449	102	6259
Região Autónoma da Madeira	853,84744897	1830,3489795	976,50153061	1220,2326530
	9592	9184	2245	6122
Região Autónoma dos Açores	828,65816326	1698,5938775	869,93571428	1132,3959183
	5306	5102	5714	6735
Região de Aveiro	1267,2045918	2664,1408163	1396,9362244	1776,0938775
	3673	2653	898	5102
Região de Coimbra	1539,2295918	3274,5352040	1735,3056122	2183,0234693
	3673	8163	449	8776
Região de Leiria	995,32346938	2073,6367346	1078,3132653	1382,4244897
	7755	9388	0612	9592
Terras de Trás-os-Montes	401,22704081	832,75561224	431,52857142	555,17040816
	6327	4898	8571	3265
Viseu Dão Lafões	891,17295918	1887,8265306	996,65357142	1258,5510204
	3673	1224	8571	0816
Total	4399,7431790	9564,6971007	5058,1690530	6340,9130674
	2251	6685	7112	6444

FIGURA 9.11: Continuação do relatório gerado

A "Área Metropolitana de Lisboa" e a "Área Metropolitana do Porto" apresentam as maiores populações totais, refletindo a centralização populacional nas duas maiores cidades de Portugal. O "Continente" tem a maior população total entre todas as regiões listadas, o que é esperado já que abrange a maior parte do território continental de Portugal. Em muitas regiões, há uma distribuição relativamente equilibrada entre homens e mulheres, embora algumas áreas apresentem pequenas variações. Por exemplo, a região do Algarve e o Alto Alentejo mostram uma proporção um pouco maior de mulheres em comparação com homens. Regiões como "Baixo Alentejo", "Beiras e Serra da Estrela" e "Medio Tejo" têm populações menores. Isto pode indicar áreas com menor densidade populacional ou possivelmente áreas rurais com menos atração urbana.

Capítulo 10

Rownumber, rank, denserank e partition

10.1 Aplicar o rownumber, rank, denserank e partition

The screenshot shows a SQL query in a text editor. The query selects several columns from a table named 'dados' and adds three calculated columns using window functions: ROW_NUMBER(), RANK(), and DENSE_RANK(). All three functions are partitioned by 'id_condicaotrab' and ordered by 'valor' in descending order.

```

SELECT
    id_dados,
    id_ano,
    valor,
    id_localre,
    id_sexo,
    id_grupoetario,
    id_condicaotrab,
    id_nivelescolaridade,
    ROW_NUMBER() OVER (PARTITION BY id_condicaotrab ORDER BY valor DESC) AS RowNumber,
    RANK() OVER (PARTITION BY id_condicaotrab ORDER BY valor DESC) AS Rank,
    DENSE_RANK() OVER (PARTITION BY id_condicaotrab ORDER BY valor DESC) AS DenseRank
FROM dados;

```

Below the query, the results are displayed in a table with 11 columns: id_dados, id_ano, valor, id_localre, id_sexo, id_grupoetario, id_condicaotrab, id_nivelescolaridade, RowNumber, Rank, and DenseRank. The results show 6 rows of data.

	id_dados	id_ano	valor	id_localre	id_sexo	id_grupoetario	id_condicaotrab	id_nivelescolaridade	RowNumber	Rank	DenseRank
1	392984	S7A2021	3605	20	T	08	1	22	8101	8101	5971
2	231566	S7A2021	3605	112	2	03	1	3	8102	8101	5971
3	387104	S7A2021	3605	2	T	08	1	22	8103	8101	5971
4	112142	S7A2011	3605	161	1	04	1	2	8104	8101	5971
5	63704	S7A2011	3602	15	T	08	1	22	8105	8105	5972
6	69584	S7A2011	3602	150	T	08	1	22	8106	8105	5972

FIGURA 10.1: Rownumber, rank, denserank e partition

O código SQL realiza uma consulta na tabela "dados", selecionando várias colunas, incluindo id dados, id ano, valor, id localre, id sexo, id grupoetario, id condicaotrab e id nivelescolaridade. Além dessas colunas, são calculados três valores adicionais usando funções de janela: ROW NUMBER(), que gera um número sequencial único para cada linha dentro de cada partição do conjunto de resultados, ordenado por valor em ordem decrescente; RANK(), que atribui uma classificação a cada linha dentro de cada partição do conjunto de resultados, ordenado por valor em ordem decrescente, com valores iguais recebendo a mesma classificação; e DENSE RANK(), que também atribui uma classificação a cada linha dentro de cada partição do conjunto de resultados, ordenado por valor em ordem decrescente, mas sem pular números em caso de valores empatados. Essas funções de janela são particionadas pela coluna id condicaotrab e ordenadas pela coluna valor em ordem decrescente, resultando em um conjunto de resultados que inclui todas as colunas originais, além das colunas calculadas RowNumber, Rank e DenseRank.

Capítulo 11

Database Maintenance

11.1 Database Maintenance

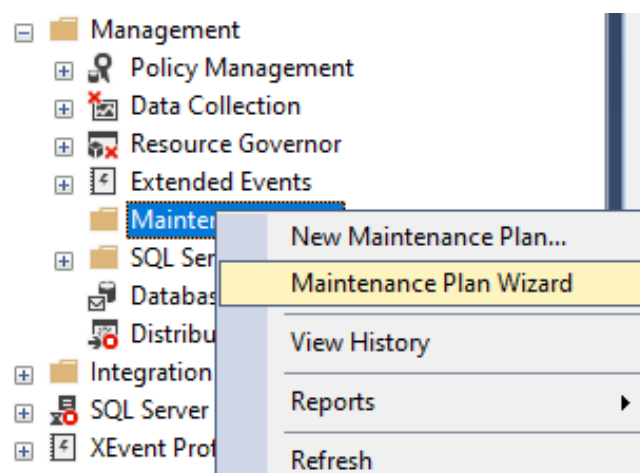


FIGURA 11.1: Abrir Maintenance Plan Wizard

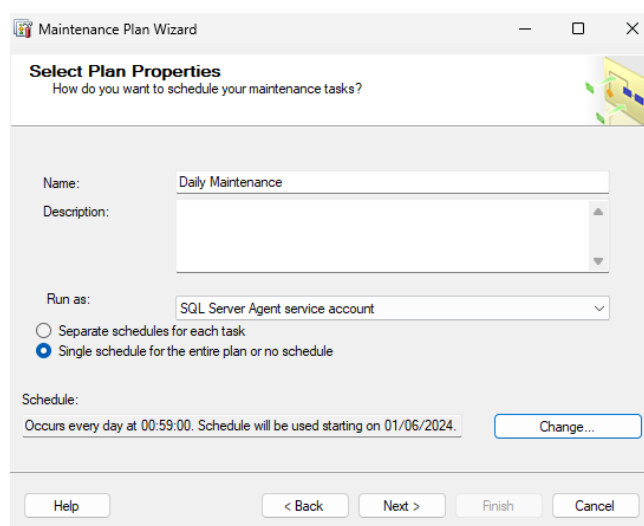


FIGURA 11.2: Mudar horário

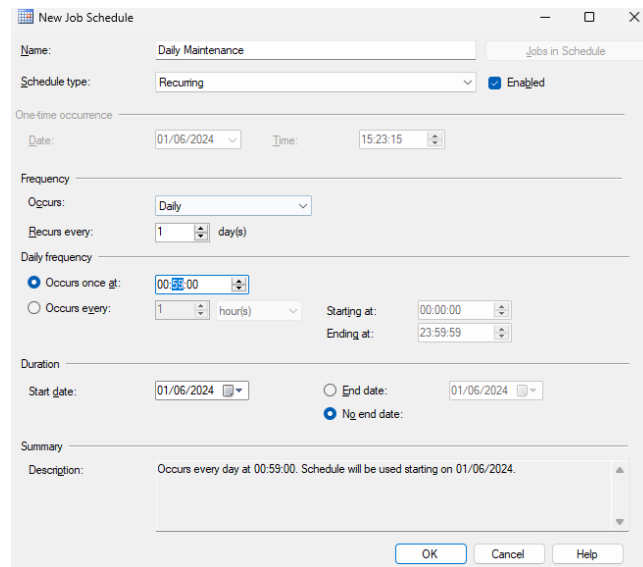


FIGURA 11.3: Alterar frequência e hora

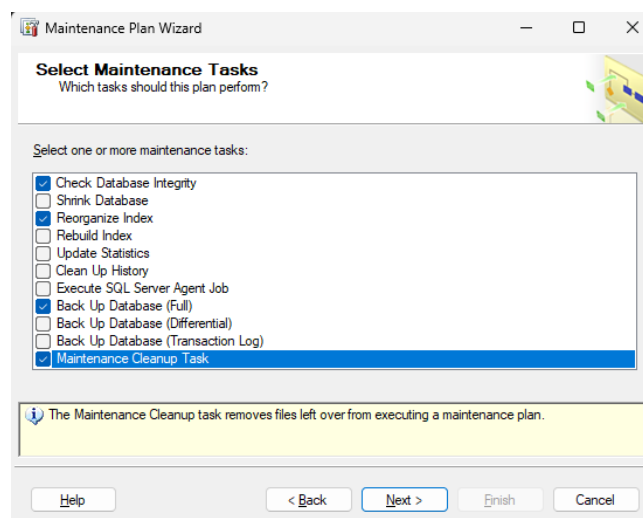


FIGURA 11.4: Selecionar tarefas de manutenção

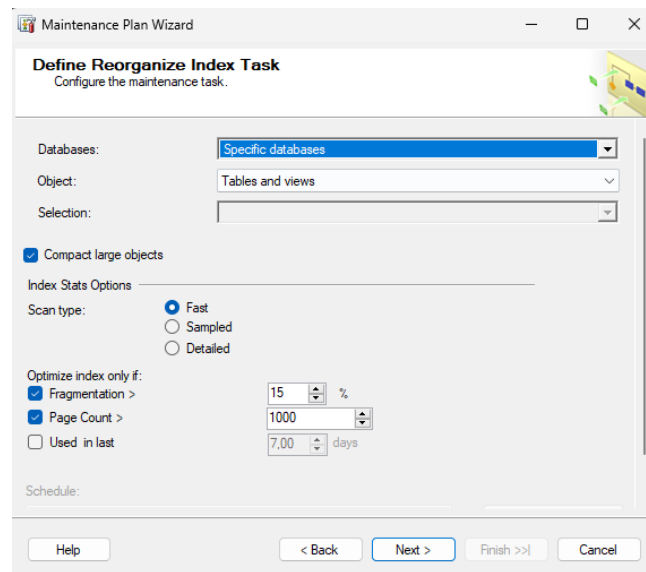


FIGURA 11.5: Especificar a base de dados

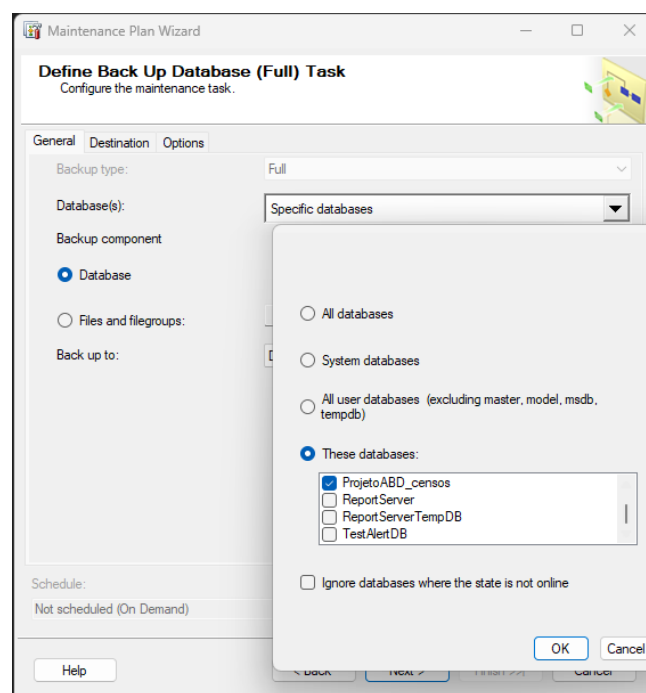


FIGURA 11.6: Especificar a base de dados

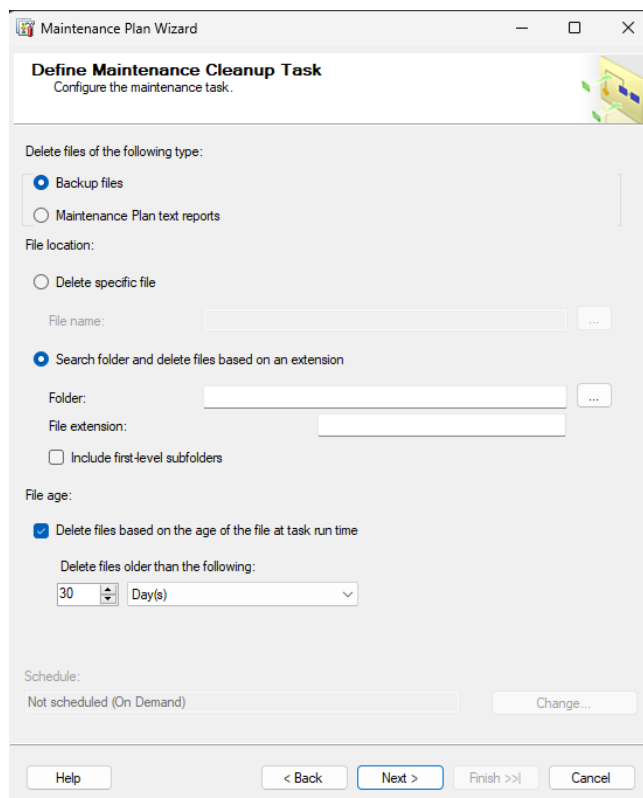


FIGURA 11.7: Definir quantidade de dias para os ficheiros serem eliminados

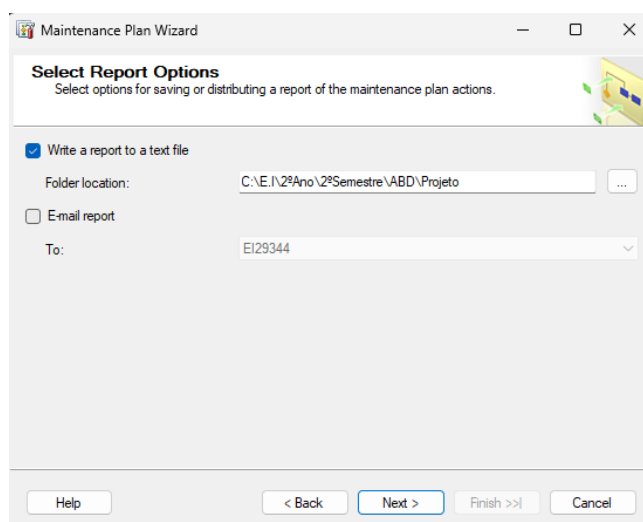


FIGURA 11.8: Escrever um relatório para um ficheiro de texto

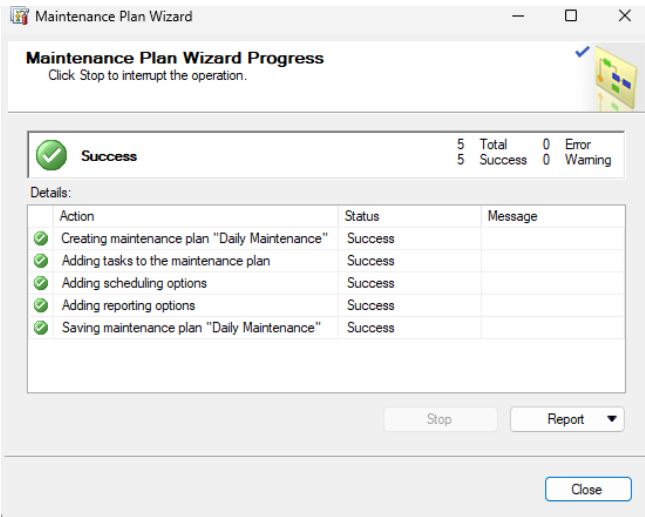


FIGURA 11.9: Plano de manutenção criado com sucesso

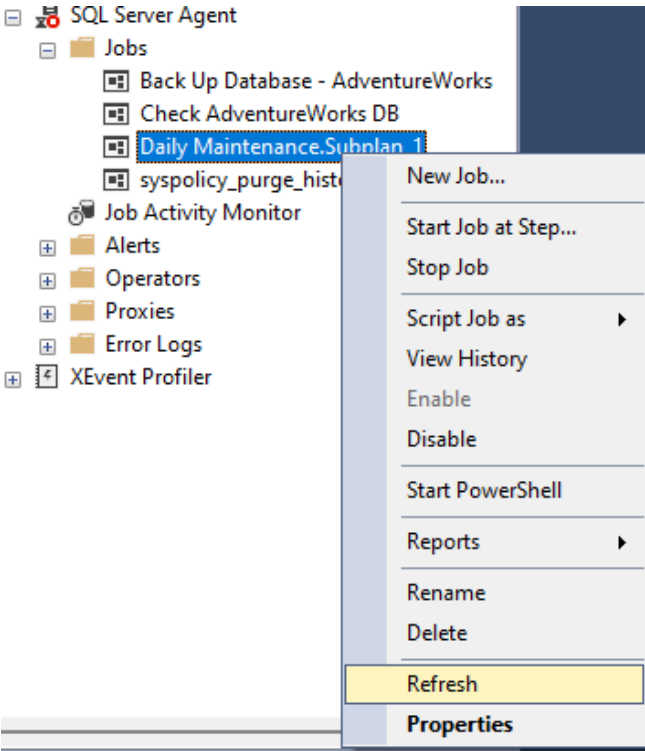


FIGURA 11.10: Propriedades do job criado

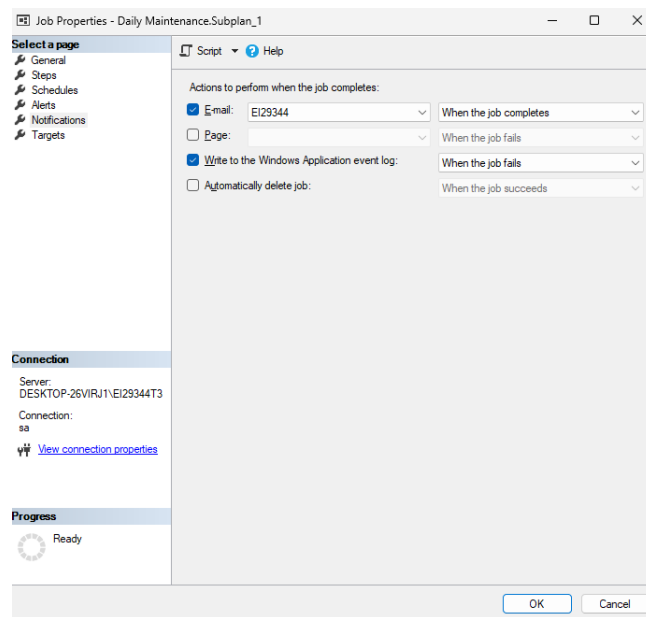


FIGURA 11.11: Enviar email quando o job for terminado

Capítulo 12

CONCLUSÕES

A realização deste trabalho prático teve como objetivo principal a aplicação dos conhecimentos adquiridos nas aulas de Administração de Bases de Dados, através da criação de um modelo de dados robusto e eficiente baseado no tema censos. Durante o desenvolvimento, foi possível explorar diversos aspectos essenciais da gestão e manipulação de dados em grande escala, bem como a implementação de mecanismos de integridade referencial e a utilização de ferramentas avançadas de SQL.

A criação do modelo entidade-relação foi a base para o desenvolvimento de uma base de dados que suporta todos os requisitos apresentados. A implementação incluiu a criação de tabelas, funções, views complexas, stored procedures, triggers, e a utilização de cursors para operações específicas. Adicionalmente, foram implementados mecanismos de logging e histórico para garantir a rastreabilidade das operações realizadas na base de dados.

A análise de desempenho com o Execution Plan e o Database Engine Tuning Advisor possibilitou a otimização das instruções SQL, assegurando que a base de dados opere de maneira eficiente mesmo sob cargas de trabalho intensas. Além disso, a implementação de rotinas de manutenção de dados e índices, bem como a configuração de backups diários, assegurou a integridade e a continuidade do sistema de base de dados.

A elaboração de relatórios com SSRS, incluindo gráficos e tabelas, permitiu a visualização e análise dos dados de forma clara e concisa, facilitando a tomada de decisões baseada em estatísticas precisas e atualizadas.

Este trabalho não só consolidou os conhecimentos teóricos, mas também proporcionou uma experiência prática valiosa na administração de bases de dados, destacando a importância de um design bem elaborado e a implementação cuidadosa de todas as funcionalidades necessárias para garantir a eficiência, integridade e segurança dos dados.