



ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO

Disciplina de Bases de Dados

Ano Letivo de 2023/2024

Gestão de Estufas

8220229 | Miguel Tavares

8220805 | Francisco Costa

8220191 | Bruno Ferreira

Maio, 2024

Data de Receção	
Responsável	
Avaliação	
Observações	

Gestão de Estufas

8220229 | Miguel Tavares

8220805 | Francisco Costa

8220191 | Bruno Ferreira

Maio. 2024

Dedicatória

Gostaríamos de expressar a nossa gratidão ao professor de Base de Dados pelo suporte fornecido ao longo deste projeto e pelos conhecimentos transmitidos, que foram fundamentais para o desenvolvimento bem-sucedido deste trabalho.

Resumo

O Trabalho Prático "Gestão de Estufas" foi proposto pela Unidade Curricular de Bases de Dados da Escola Superior de Tecnologia e Gestão para o ano letivo de 2023/2024. O objetivo é criar uma base de dados para informatizar a gestão de estufas de uma empresa, facilitando o registo e consulta de informações sobre plantas, legumes e frutas.

O projeto envolve:

Coleta, armazenamento e organização de dados necessários para a construção da base de dados.

Modelo Conceitual: Foco nas entidades e seus relacionamentos, conforme a metodologia de desenho de bases de dados ensinada nas aulas.

Modelo Lógico: Apresentação do modelo lógico, com validações e verificações usando a técnica de normalização.

Modelo Físico: Implementação da base de dados final no Microsoft SQL Server, incluindo tipos de dados, triggers, stored procedures, functions e views.

Os dados a serem registados incluem detalhes sobre a plantação e colheita, quantidade de sementes e consumo de produtos auxiliares. Consultas específicas devem ser atendidas, como produtos colhidos no mês anterior e quantidade semeada por trimestre.

Conteúdo

Resumo.....	ii
Índice de Figuras	iv
Índice de Tabelas.....	vi
1. Introdução	1
1.1 Contextualização	1
1.2 Apresentação do Caso de Estudo	1
1.3 Motivação e Objetivos	3
1.4 Estrutura do Relatório	3
2. Construção do Modelo Concetual.....	4
2.1 Identificação das entidades	4
2.2 Identificação de relacionamentos e respetiva multiplicidade	5
2.3 Identificação e associação de atributos a entidades ou relacionamentos	6
2.4 Identificação dos domínios de atributos	10
2.5 Escolha das chaves candidatas, primárias e alternativas	11
2.6 Verificação de redundância no modelo	12
2.7 Validação das transações.....	13
2.8 Validação do modelo com o utilizador.....	13
3. Construção do modelo lógico	14
3.1 Derivação de relações para modelo de dados lógico.....	14
3.2 Modelo de dados lógico resultante.....	18
3.3 Validar modelo lógico com Normalização.....	19
Modelo de dados lógico no final da normalização	28
3.4 Validar novamente o modelo com transações	29
3.5 Identificar Restrições de Integridade	29
3.6 Regras de negócio	32
3.7 Validação do modelo com o utilizador.....	33
3.8 Verificação do provável crescimento futuro	33
4. Construção do modelo físico.....	34
4.1 Criação das tabelas no SQL Server Management Studio	35
4.2 Criação das procedures de CUD (Create, Update, Delete) no SQL Server Management Studio	42
4.3 Consultas propostas	48
4.3 Triggers.....	50
4.3 SQL SERVER AGENT	56
5. Conclusões e Trabalho Futuro.....	58

Bibliografia	59
Referências WWW	60
Lista de Siglas e Acrônimos	61

Índice de Figuras

Figura 1 - Primeiro esboço do Modelo Concetual.....	6
Figura 2 - Esboço do modelo concetual com chaves primárias	11
Figura 3 - Modelo lógico de dados com todos os atributos	18
Figura 4 - Modelo logico de dados lógico depois da normalização	28
Figura 5 - Diagrama ER do modelo fisico noSQL	34
Figura 6 - Create Table Estufa	35
Figura 7 - Create Table Secção	36
Figura 8 - Create Table Tipo	37
Figura 9 - Create Table Produto.....	37
Figura 10 - Create Table Plantio	38
Figura 11 - Create Table Plantio Consumivel	39
Figura 12 - Create Table Consumivel	40
Figura 13 - Create Table Colheita	40
Figura 14 - Create Table Funcionario	41
Figura 15 - Procedure Atualizar Colheita	42
Figura 16 - Procedure Eliminar Colheita	43
Figura 17 - Procedure Inserir Colheita	43
Figura 18 - Procedure Atualizar Estufa	44
Figura 19 - Procedure Eliminar Estufa	44
Figura 20 - Procedure Inserir Estufa	45
Figura 21 - Procedure Atualizar Plantio	46
Figura 22 - Procedure Eliminar Plantio	46
Figura 23 - Procedure Inserir Plantio	47
Figura 24 - Procedure Reverter Plantio	47
Figura 25 - Consulta ' Quais os produtos colhidos no mês anterior'	48
Figura 26 - Resultado exemplo da consulta ' Quais os produtos colhidos no mês anterior'	48
Figura 27 - Consulta ' Qual a quantidade semeada por trimestre por produto no último ano' ...	48

Figura 28 - Resultado exemplo ' Qual a quantidade semeada por trimestre por produto no último ano'	48
Figura 29 - Consulta 'Lista dos produtos com maior quantidade colhida no ano de 2018'	48
Figura 30 - Resultado exemplo da consulta lista dos produtos com maior quantidade colhida no ano de 2018'	48
Figura 31 - Consulta 'vista geral da agricultura'	49
Figura 32 - Exemplo de resultado da consulta 'vista geral agricultura'	49
Figura 33 - trigger 'verificar area da estufa'	50
Figura 34 - Trigger 'limite secções estufa'	51
Figura 35 - Trigger Verificar Tipo	52
Figura 36 - Trigger Limite Produto / Secção	53
Figura 37 - Trigger Limite Plantações Por Secção	54
Figura 38 - Trigger Atualizar Data Fim Plantio	55
Figura 39-Step do SQL SERVER AGENT	56
Figura 40-Agendamento para execução do Job	57

Índice de Tabelas

Tabela 1 - Identificação das entidades	4
Tabela 2 - Identificação de relacionamentos e respetiva multiplicidade	5
Tabela 3 - Identificação e associassão de atributos da entidade Estufa	6
Tabela 4 - Identificação e associassão de atributos da entidade Secção	7
Tabela 5 - Identificação e associassão de atributos da entidade Tipo	7
Tabela 6 - Identificação e associassão de atributos da entidade Produto	7
Tabela 7 - Identificação e associassão de atributos da entidade Plantio	8
Tabela 8 - Identificação e associassão de atributos da entidade Colheita	8
Tabela 9 - Identificação e associassão de atributos da entidade Consumível	9
Tabela 10 - Identificação e associassão de atributos da entidade Funcionário	9
Tabela 11 - Domínios de atributos	10
Tabela 12 - Identificação das chaves primárias e chaves candidatas	11
Tabela 13 - Documento registo de um funcionário	19
Tabela 14 - Forma não normalizada – Registo Funcionário	20
Tabela 15 - 1ª forma normalizada - Registo de funcionário	20
Tabela 16 - Documento do relatório da plantação	22
Tabela 17 - Forma não normalizada - relatório plantio	23
Tabela 18 - 1ª forma normalizada - relatório plantio	23
Tabela 19 - Documento relatório colheitas de uma plantação	25
Tabela 20 - Forma não normalizada - relatório colheitas de uma plantação	26
Tabela 21 - 1ª forma normalizada - Relatório colheitas de uma plantação	26
Tabela 22 - Entidade Estufa e as suas restrições de integridade	29
Tabela 23 - Entidade Secção e as suas restrições de integridade	29
Tabela 24 - Entidade Tipo e as suas restrições de integridade	30
Tabela 25 - Entidade Produto e as suas restrições de integridade	30
Tabela 26 - Entidade Plantio e as suas restrições de integridade	30
Tabela 27 - Entidade Colheita e as suas restrições de integridade	31
Tabela 28 - Entidade Consumível e as suas restrições de integridade	31
Tabela 29 - Entidade Plantio_Consumível e as suas restrições de integridade	31
Tabela 30 - Entidade Funcionário e as suas restrições de integridade	32

1. Introdução

1.1 Contextualização

O presente projeto consiste no desenvolvimento de uma base de dados para a gestão de estufas de uma empresa, conforme proposto aos alunos da unidade curricular de Bases de Dados. A necessidade principal é informatizar os processos de registo e consulta de informações sobre plantas, legumes e frutas, otimizando a gestão e armazenamento de dados.

Inicialmente, será realizada uma análise detalhada dos requisitos fornecidos. Em seguida, seguindo a metodologia de desenho de Bases de Dados abordada nas aulas, será elaborado o modelo conceitual, que servirá como suporte inicial do projeto, focando nas entidades e seus relacionamentos.

Após a elaboração do modelo conceitual, será desenvolvido o modelo lógico, que incluirá um conjunto de validações e verificações utilizando a técnica de normalização aprendida em sala de aula. Finalmente, será apresentado o modelo físico, que consiste na implementação final da base de dados. Para isso, será utilizado o Microsoft SQL Server, incluindo a criação de tipos de dados, triggers, stored procedures, functions e views necessárias para atender aos requisitos e consultas propostas.

1.2 Apresentação do Caso de Estudo

Uma empresa especializada em gestão agrícola contactou a ESTG com um desafio para os alunos da unidade curricular de Bases de Dados. O desafio consiste no desenvolvimento de uma base de dados que armazenará todas as informações relacionadas à gestão de estufas, com o

objetivo de informatizar e otimizar os processos de registo e consulta de dados sobre plantas, legumes e frutas. As especificações básicas são as seguintes:

Organização das Estufas:

- As estufas são divididas em secções: Plantas Aromáticas, Legumes e Frutas. Cada estufa pode ter no máximo 3 secções
- Cada secção pode conter até 10 produtos diferentes, ou seja, cada secção só pode ter um número de plantações indefinido, mas no máximo só pode plantar 10 produtos diferentes no mesmo intervalo de tempo

Registo de Plantação e Colheita:

- Registrar informações sobre a plantação e colheita dos produtos.
- Incluir detalhes sobre a quantidade de sementes utilizadas e o consumo de produtos auxiliares (adubo, água, pesticidas, etc.).

Para guiar a construção do protótipo da base de dados, foram formuladas algumas questões de consulta para garantir que o sistema possa responder adequadamente às necessidades. As consultas são as seguintes:

- Identificar todos os produtos que foram colhidos no mês anterior.
- Listar a quantidade de sementes utilizadas por produto em cada trimestre do último ano.
- Identificar quais produtos tiveram a maior quantidade colhida no ano de 2018.

1.3 Motivação e Objetivos

Aos alunos da unidade curricular de Bases de Dados foi proposto pelos docentes responsáveis o desenvolvimento de um projeto voltado para a criação de uma base de dados para a gestão de estufas. Este projeto permite aplicar na prática os conhecimentos adquiridos em sala de aula, consolidando o aprendizado e contribuindo para o desenvolvimento de competências essenciais para o futuro profissional.

Para atingir os objetivos do projeto, consideramos os requisitos apresentados no enunciado. Foi realizado um planejamento meticuloso dos métodos a serem utilizados, visando a elaboração de uma base de dados eficiente e robusta. Pretendemos que o projeto resulte em uma solução que satisfaça plenamente as necessidades de gestão de uma empresa de estufas, conforme especificado no enunciado e com base em cenários reais.

1.4 Estrutura do Relatório

Durante os próximos capítulos deste relatório, serão apresentadas as fases da metodologia de desenho de bases de dados, conforme abordadas nas aulas, incluindo os modelos conceitual, lógico e físico. Serão detalhadas as entidades, seus atributos, relacionamentos, multiplicidade, domínio, chaves primárias e candidatas, além de consultas e triggers para o melhor funcionamento da base de dados.

2. Construção do Modelo Concetual

2.1 Identificação das entidades

Dado o problema proposto encontramos as seguintes entidades:

Entidade	Descrição	Apelido	Ocorrência
Estufa	Uma estrutura projetada para o cultivo e proteção de plantas em condições ambientais controladas.		Pode haver várias estufas em uma empresa, então podemos considerar uma ocorrência múltipla.
Secção	Segmento ou área específica designada para um tipo específico de planta ou processo dentro de uma estrutura ou área maior.		Cada estufa pode conter várias secções para diferentes tipos de plantas, então podemos ter várias secções por estufa. Máximo 3.
Tipo	Categoria que define a natureza de um produto ou secção, como Plantas Aromáticas, Legumes, Frutas, etc.		Cada secção e cada produto pertencem a um único tipo.
Produto	Itens que são produzidos a partir da plantação, incluindo, frutas, vegetais, flores, etc.		Cada secção pode cultivar até 10 produtos diferentes ao mesmo tempo.
Plantio	Uma área de terra onde as plantas são cultivadas em grande escala.		Cada produto pode ser plantado em uma área específica, então podemos ter vários plantio por produto
Consumível	Produtos consumíveis diretamente, associados aos plantios.		Cada consumível pode estar associado a vários plantios, então podemos ter vários consumíveis por plantio.
Colheita	O processo ou período de coleta de culturas.		Cada produto pode ser colhido uma vez, então podemos ter uma colheita por produto.
Funcionário	Pessoa responsável pelo trabalho nas estufas.		Cada funcionário trabalha em uma única secção, mas uma secção pode ter vários funcionários.

Tabela 1 - Identificação das entidades

2.2 Identificação de relacionamentos e respetiva multiplicidade

Entity Name	Multiplicity	Relationship	Multiplicity	Entity Name
Estufa	1..1	possui	1..3	Secção
Secção	1..3	Pertence a	1..1	Estufa
	1..1	contém	0..*	Plantio
	1..*	tem	1..1	Tipo
	1..1	contém	1..*	Funcionario
Tipo	1..1	pertence	1..*	Secção
	1..1	tem	1..*	Produto
Plantio	0..*	utiliza	0..*	Consumível
	1..1	resulta	0..*	colheita
	1..*	forma	1..1	produto
	0..*	Pertence a	1..1	Secção
Produto	1..1	É formado	1..*	Plantio
	1..*	pertence	1..1	Tipo
colheita	0..*	associada	1..1	plantio
Consumível	0..*	É requerido	0..*	plantio
Funcionário	1..*	Trabalha em	1..1	Secção

Tabela 2 - Identificação de relacionamentos e respetiva multiplicidade

De maneira que seja mais perceptível a identificação dos relacionamentos e respectiva multiplicidade é representado um primeiro esboço do diagrama do modelo concetual.

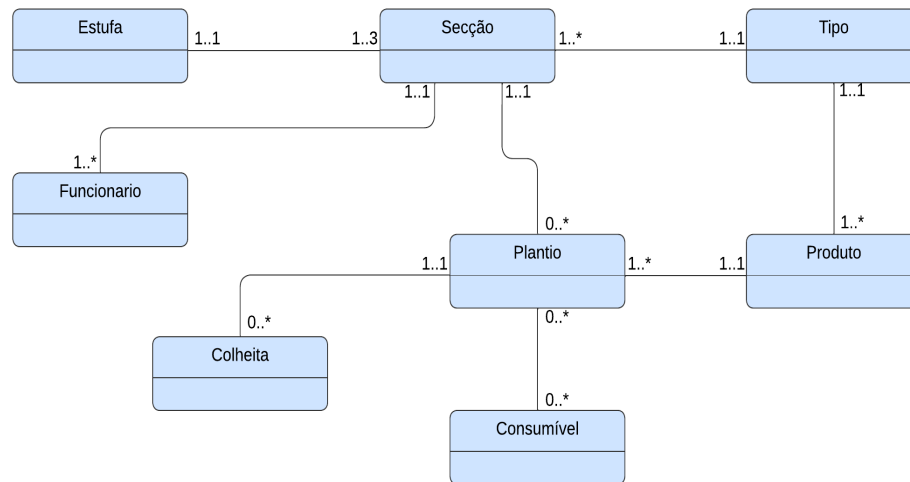


Figura 1 - Primeiro esboço do Modelo Conceitual

2.3 Identificação e associação de atributos a entidades ou relacionamentos

Depois de encontradas as entidades e suas relações chegamos ao ponto da identificação dos atributos para a respectiva entidade

Entidade	Atributo	Descrição	Domínio	Null	Multivalores
Estufa	Id_Estufa	Identificador único da estufa	idType	Não	Não
	Nome	Nome da estufa	nameType	Não	Não
	Localização	Localização geográfica da estufa	nameType	Não	Não
	area	Área total da estufa em metros quadrados	areaType	Não	Não
	Temperature	Faixa de temperatura controlada dentro da estufa	MedidaType	Não	Não
	Umidade	Nível de umidade controlado dentro da estufa	MedidaType	Não	Não

Tabela 3 - Identificação e associassão de atributos da entidade Estufa

Entidade	Atributo	Descrição	Domínio	Null	Multivalores
Secção	Id_Seccao	Identificador único da secção	idType	Não	Não
	Descrição	Breve descrição do tipo de plantas cultivadas na secção	descricaoType	Não	Não
	area	Área da secção em metros quadrados	areaType	Não	Não

Tabela 4 - Identificação e associassão de atributos da entidade Secção

Entidade	Atributo	Descrição	Domínio	Null	Multivalores
Tipo	Id_Tipo	Identificador único do tipo	idType	Não	Não
	nome	Nome do tipo do produto/secção	nomeType	Não	Não

Tabela 5 - Identificação e associassão de atributos da entidade Tipo

Entidade	Atributo	Descrição	Domínio	Null	Multivalores
Produto	Id_Produto	Identificador único da produto	idType	Não	Não
	Nome	Nome do produto	nameType	Não	Não
	Decricao	Descrição detalhada do produto	descricaoType	Não	Não
	duracao	Duração em dias que o produto demora a ser produzido	numType	Não	Não

Tabela 6 - Identificação e associassão de atributos da entidade Produto

Entidade	Atributo	Descrição	Domínio	Null	Multivalores
Plantio	Id_Plantio	Identificador único do plantio	idType	Não	Não
	dataInicioPlantio	Data do início da plantação	dataType	Não	Não
	dataFimPlantio	Data do fim da plantação	dataType	Sim	Não
	quantidadeSementes	Quantidade de sementes plantadas	numType	Não	Não
	Colhido	Indica se a plantação já foi totalmente colhida	booleanType	Não	Não
	emUso	Indica se a plantação está em uso	booleanType	Não	Não

Tabela 7- Identificação e associassão de atributos da entidade Plantio

Entidade	Atributo	Descrição	Domínio	Null	Multivalores
Colheita	Id_Colheita	Identificador único da colheita	idType	Não	Não
	dataColheita	Data da colheita	dataType	Não	Não
	quantidadeColhida	Quantidade de produto colhido	numType	Não	Não
	observações	Observação da colheita	descricaoType	Não	Não

Tabela 8 - Identificação e associassão de atributos da entidade Colheita

Entidade	Atributo	Descrição	Domínio	Null	Multivalores
Consumível	Id_Consumivel	Identificador único do consumível	idType	Não	Não
	Nome	Nome do consumível (ex: adubo, pesticida, água)	nameType	Não	Não
	descricao	Descrição detalhada do consumível	descricaoType	Não	Não

Tabela 9 - Identificação e associassão de atributos da entidade Consumível

Entidade	Atributo	Descrição	Domínio	Null	Multivalores
Funcionario	Id_Funcionario	Identificador único do funcionario	idType	Não	Não
	Nome	Nome do funcionario	nameType	Não	Não
	apelido	Apelido do funcionário	nameType	Não	Não
	morada	Morada funcionário	nameType	Não	Não
	dataNascimento	Data de nascimento do funcionário	dataType	Não	Não
	telemovel	Número de telemovel	telfType	Não	Não
	email	Email do funcionário	emailType	Não	Não

Tabela 10 - Identificação e associassão de atributos da entidade Funcionario

2.4 Identificação dos domínios de atributos

Domínio	DataType	Restrições
idType	Nvarchar(15)	Nenuma
nameType	Varchar(50)	Nenhuma
areaType	Decimal(5,2)	Tem de ser um valor maior que 0.
numType	Int	Tem de ser um valor positivo ou 0
medidaType	int	Nenhuma
descricaoType	Varchar(200)	Nenhuma
dataType	date	Nenhuma
boolenaType	Bit	Deve ser true(1) ou false(0) DEFAULT 0
telfType	Varchar(9)	Tem de ter 9 dígitos
emailType	Varchar(30)	Tem de conter um @

Tabela 11 - Domínios de atributos

2.5 Escolha das chaves candidatas, primárias e alternativas

Entidade	Chave Primária	Chave(s) Candidata(s)
Estufa	Id_Estufa	-
Secção	Id_Seccao	-
Tipo	Id_Tipo	Nome
Produto	Id_Produto	-
Plantio	Id_Plantio	-
Consumível	Id_Consumivel	-
Colheita	Id_Colheita	-
Funcionario	Id_Funcionario	-

Tabela 12 - Identificação das chaves primárias e chaves candidatas

Foi reconstruído o modelo concetual agora com as chaves primárias.

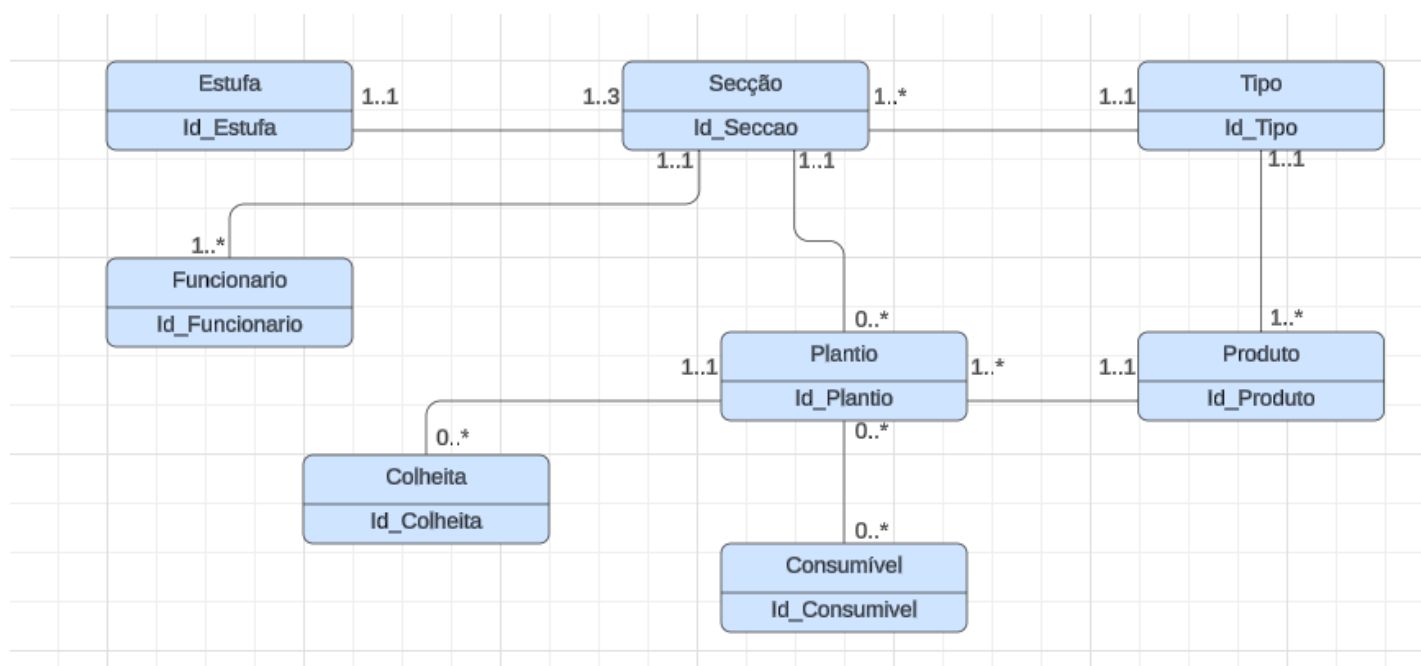


Figura 2 - Esboço do modelo concetual com chaves primárias

2.6 Verificação de redundância no modelo

O processo de verificação de redundância no modelo de dados envolve várias etapas para garantir que não haja duplicações desnecessárias e que a integridade dos dados seja mantida. Abaixo está um exemplo adaptado ao contexto do sistema de gestão de estufas:

Passo 1: Verificação de Relacionamentos 1:1

O primeiro passo desta etapa envolve verificar se existem entidades com um relacionamento 1:1. Na nossa base de dados não foram encontrados relacionamentos 1:1 entre entidades. Portanto, não há necessidade de ajustes neste aspeto do modelo.

Passo 2: Identificação de Relacionamentos Redundantes

O segundo passo consiste em identificar relacionamentos redundantes. Um relacionamento é considerado redundante quando transmite a mesma informação que outro relacionamento existente e não possui um propósito significativo adicional.

Ao revisar o diagrama de relacionamentos, identificamos que as entidades Estufa, Secção e Plantio estão inter-relacionadas. No entanto, essas relações não são redundantes, pois cada uma fornece informações únicas e específicas sobre a gestão das estufas. Por exemplo, a entidade Plantio possui dados sobre as plantações específicas dentro de uma Secção da Estufa, enquanto a Secção detalha as diferentes áreas dentro de uma Estufa.

Passo 3: Verificação do Propósito dos Relacionamentos (Dimensão Temporal)

O terceiro e último passo envolve a verificação do propósito dos relacionamentos, também conhecida como "dimensão temporal". É essencial considerar o contexto de cada relacionamento para determinar sua importância e evitar a remoção de relacionamentos aparentemente redundantes que são, na verdade, cruciais para o funcionamento do sistema.

Por exemplo, observamos que há dois tipos de relacionamentos que podem parecer redundantes: Plantio e Colheita. Embora ambos se relacionem à produção agrícola dentro das estufas, um está focado no início do ciclo de vida das plantas (plantio) e o outro na sua conclusão (colheita). Ambos são necessários para o controle eficaz da produção e a gestão dos ciclos de cultivo. A remoção de qualquer um desses relacionamentos comprometeria a capacidade de monitorar e gerenciar os processos de cultivo nas estufas.

Conclusão

Após a análise detalhada do modelo de dados, concluímos que:

Não existem relacionamentos 1:1 que necessitem de reavaliação.

Não foram identificados relacionamentos redundantes que transmitem a mesma informação.

Todos os relacionamentos existentes têm um propósito claro e são essenciais para o bom funcionamento do sistema de gestão de estufas.

Portanto, o modelo de dados atual é eficiente e bem estruturado, sem redundâncias desnecessárias, garantindo a integridade e a funcionalidade da base de dados.

2.7 Validação das transações

Para garantir que o modelo conceitual da base de dados desenvolvida para a gestão de estufas suporta as transações necessárias, verificaremos se as relações e entidades presentes são capazes de atender às consultas e operações esperadas.

Reveremos os casos de uso exemplares para demonstrar a funcionalidade do modelo em situações práticas. Verificamos a consistência e a integridade dos dados, assegurando que todas as chaves primárias estavam corretamente definidas e que as regras de negócio eram respeitadas. Além disso, consideramos a performance do sistema, garantindo que índices apropriados foram aplicados nas colunas frequentemente consultadas.

Conclusão

O modelo conceitual atual da base de dados para a gestão de estufas é capaz de suportar todas as transações mencionadas. As entidades e seus relacionamentos fornecem uma estrutura robusta para registrar, consultar e analisar dados sobre plantações, colheitas e produtos auxiliares, atendendo aos requisitos especificados no projeto.

2.8 Validação do modelo com o utilizador

Tendo por base todos os pontos anteriores é possível dizer que o modelo é válido e cumpre todas as necessidades propostas logo é possível proceder para as próximas etapas.

3. Construção do modelo lógico

3.1 Derivação de relações para modelo de dados lógico

Neste passo, vamos derivar os relacionamentos para o modelo lógico com base no modelo conceitual apresentado. Esta derivação inclui a definição de entidades, atributos, chaves primárias, chaves estrangeiras e a tradução de relacionamentos conforme identificado no modelo conceitual.

Entidades Fortes

As entidades fortes são aquelas cuja existência não depende de outras entidades. Abaixo estão as entidades fortes identificadas, já com as chaves estrangeiras provenientes das outras entidades:

1. **Estufa** (Estufa)
 - Atributos: **Id_Estufa (PK)**, **Nome**, **Localizacao**, **Area**, **Temperatura**, **Umidade**
 - Chave Primária: **Id_Estufa**
2. **Secção** (Secao)
 - Atributos: **Id_Seccao (PK)**, **Descricao**, **Area**, **Id_Estufa (FK)**, **Id_Tipo(FK)**
 - Chave Primária: **Id_Seccao**
 - Chave Estrangeira: **Id_Estufa** referencia **estufa (Id_Estufa)**
 - Chave Estrangeira: **Id_Tipo** referencia **Tipo (Id_Tipo)**
3. **Tipo** (Tipo)
 - Atributos: **Id_Tipo (PK)**, **Nome**
 - Chave Primária: **Id_Tipo**
4. **Produto** (Produto)
 - Atributos: **Id_Produto (PK)**, **Nome**, **Descricao**, **Duracao**, **Id_Tipo (FK)**
 - Chave Primária: **Id_Produto**
 - Chave Estrangeira: **Id_Tipo** referencia **Tipo (Id_Tipo)**
5. **Plantio** (Plantio)
 - Atributos: **Id_Plantio (PK)**, **DataInicioPlantio**, **DataFimPlantio**, **QuantidadeSementes**, **Colhido**, **EmUso**, **Id_Secao (FK)**, **Id_Produto (FK)**
 - Chave Primária: **Id_Plantio**
 - Chave Estrangeira: **Id_Seccao** referencia **Seccao(Id_Seccao)**
 - Chave Estrangeira: **Id_Produto** referencia **Produto (Id_Produto)**

6. **Colheita** (Colheita)

- Atributos: **Id_Colheita(PK)**, **DataColheita**, **QuantidadeColhida**, **Observacoes**, **Id_Plantio (FK)**
- Chave Primária: **Id_Colheita**
- Chave Estrangeira: **Id_Plantio** referencia **Plantio(Id_Plantio)**

7. **Consumivel** (Consumivel)

- Atributos: **Id_Consumivel (PK)**, **Nome**, **Descricao**
- Chave Primária: **Id_Consumivel**

8. **Funcionario** (Funcionario)

- Atributos: **Id_Funcionario (PK)**, **Nome**, **Apelido**, **Morada**, **DataNascimento**, **Telemovel**, **Email**, **Id_Seccao (FK)**
- Chave Primária: **Id_Funcionario**
- Chave Estrangeira: **Id_Seccao** referencia **Seccao(Id_Seccao)**

Entidades Fracas

Tendo em conta todos os esboços apresentados anteriormente do modelo concetual podemos concluir que não existem entidades fracas

Nos próximos passos vamos proceder à identificação e avaliação dos vários tipos de relacionamentos existentes. Teremos a entidade pai e a entidade filho. À entidade filho será adicionada uma cópia da chave primária da entidade pai.

Relacionamentos 1 para Muitos (1...*)

I. Estufa 1... Secção*

- **Entidade Pai:** Estufa (Id_Estufa, Nome, Localização, Área, Temperatura, Umidade)
 - **Chave Primária:** Id_Estufa
- **Entidade Filho:** Secção (Id_Seccao, Descrição, Área, Id_Estufa)
 - **Chave Primária:** Id_Seccao
 - **Chave Estrangeira:** Id_Estufa

II. Tipo 1... Secção*

- **Entidade Pai:** Tipo (Id_Tipo, nome)
 - **Chave Primária:** Id_Tipo
- **Entidade Filho:** Secção (Id_Seccao, Descrição, Área, Id_Tipo)
 - **Chave Primária:** Id_Seccao
 - **Chave Estrangeira:** Id_Tipo

III. Secção 1... Plantio*

- **Entidade Pai:** Secção (Id_Seccao, Descrição, Área)
 - **Chave Primária:** Id_Seccao
- **Entidade Filho:** Plantio (Id_Plantio, dataInicioPlantio, dataFimPlantio, quantidadeSementes, Colhido, emUso, Id_Seccao)
 - **Chave Primária:** ID_Plantio
 - **Chave Estrangeira:** ID_Seccao

IV. Plantio 1... Colheita*

- **Entidade Pai:** Plantio (Id_Plantio, dataInicioPlantio, dataFimPlantio, quantidadeSementes, Colhido, emUso)
 - **Chave Primária:** Id_Plantio
- **Entidade Filho:** Colheita (Id_Colheita, dataColheita, quantidadeColhida, Observações, Id_plantio)
 - **Chave Primária:** Id_Colheita
 - **Chave Estrangeira:** Id_Plantio

V. Produto 1... Plantio*

- **Entidade Pai:** Produto (Id_Produto, Nome, Descrição, duração)
 - **Chave Primária:** Id_Produto
- **Entidade Filho:** Plantio (Id_Plantio, dataInicioPlantio, dataFimPlantio, quantidadeSementes, Colhido, emUso, Id_Produto)
 - **Chave Primária:** Id_Plantio
 - **Chave Estrangeira:** Id_Produto

VI. Tipo 1... Produto*

- **Entidade Pai:** Tipo (Id_Tipo, nome)
 - **Chave Primária:** Id_Tipo
- **Entidade Filho:** Produto (Id_Produto, Nome, Descrição, duração, Id_Tipo)
 - **Chave Primária:** Id_Produto
 - **Chave Estrangeira:** Id_Tipo

VII. Secção 1... Funcionário*

- **Entidade Pai:** Secção (Id_Secção, Descrição, Área)
 - **Chave Primária:** Id_Secção
- **Entidade Filho:** Funcionário (Id_Funcionario, nome, apelido, morada, dataNascimento, telemóvel, email, Id_Seccao)
 - **Chave Primária:** Id_Funcionario
 - **Chave Estrangeira:** Id_Seccao

Relacionamentos Muitos para Muitos (...)

I. Plantio ... Consumível

- **Entidade Pai 1:** Plantio (Id_Plantio, dataInicioPlantio, dataFimPlantio, quantidadeSementes, Colhido, emUso)
 - **Chave Primária:** Id_Plantio
- **Entidade Pai 2:** Consumível (Id_Consumível, Nome, Descrição)
 - **Chave Primária:** Id_Consumivel
- **Entidade Filho:** Plantio_Consumível (Id_Plantio_Consumível, Id_Plantio, Id_Consumivel, quantidade)
 - **Chave Primária:** Id_Plantio_Consumivel
 - **Chave Estrangeiras:** Id_Plantio, Id_Consumivel

3.2 Modelo de dados lógico resultante

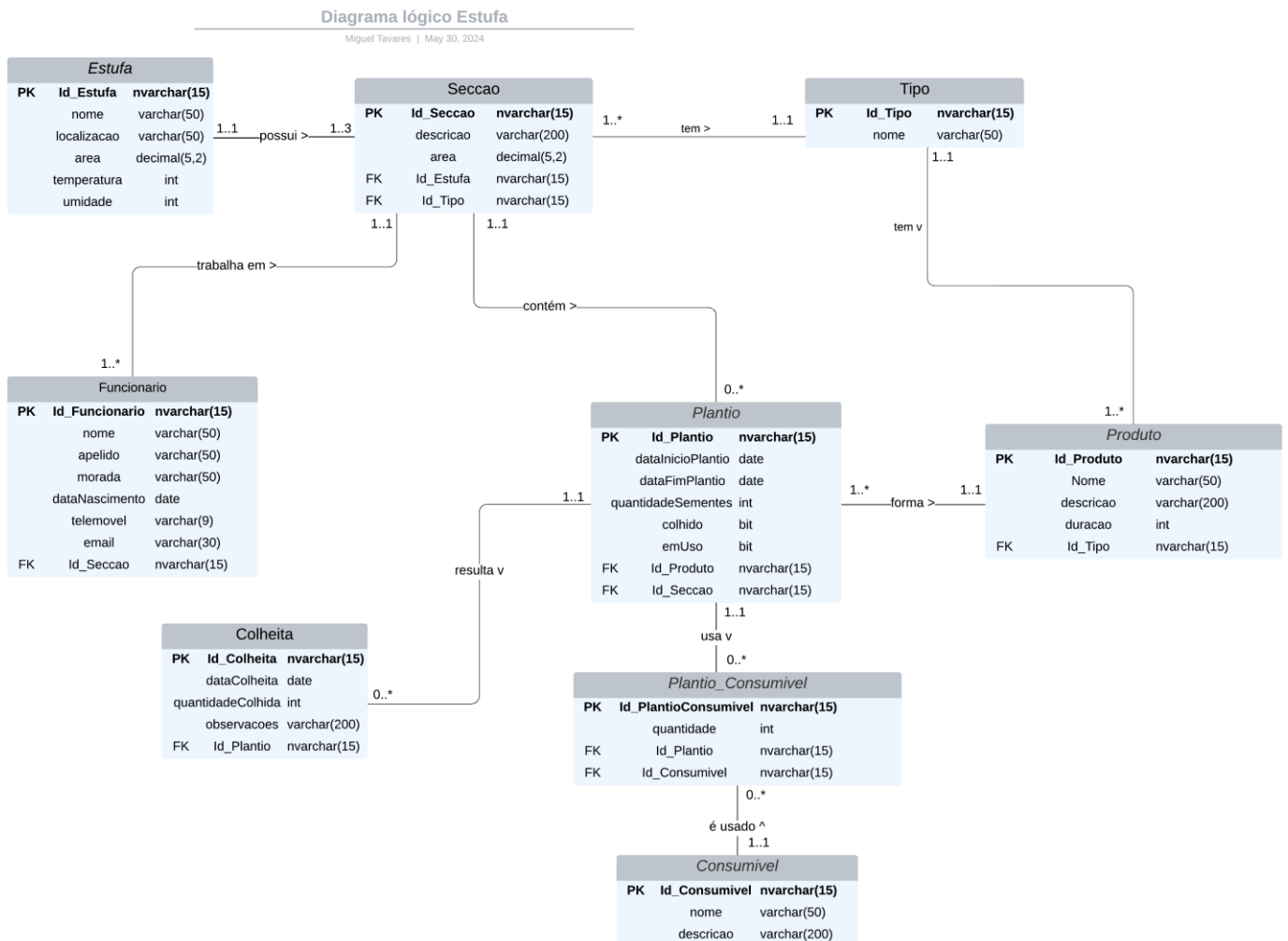


Figura 3 - Modelo lógico de dados com todos os atributos

3.3 Validar modelo lógico com Normalização

Documento registo de um funcionário na secção onde o mesmo vai trabalhar		
Informação Estufa		
Id_Estufa: E1	NomeE: Estufa1	Localização: Porto
Area: 200	Temperatura: 30	Umidade: 10
Informação Secção		
Id_Seccao: S1	Descrição: secção das frutas	Area: 50
Informação funcionário		
Id_Funcionario: F200	NomeF: Rui	
Apelido: Fernandes	Morada: Rua da base de dados	
Data Nascimento: 2/3/2000	Telemovel: 912345678	
Email: ruif@gmail.com		

Tabela 13 - Documento registo de um funcionário

Passos de Normalização

Id_Estufa – A

NomeE – B

Localizacao - C

Area – D

Temperatura – E

Umidade – F

Id_Seccao – G

Descricao – H

Area – I

Id_Funcionario – J

NomeF – K

Apelido – L

Morada – M

Data Nascimento – N

Telemovel – O

Email – P

Forma Não Normalizada

Forma inicial do documento onde há redundância e possíveis grupos repetidos. Vamos normalizar isso passo a passo.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
E1	Estufa1	Porto	200	30	10	S1	Secção das frutas	50	F200	Rui	Fernandes	Rua da base de dados	2/3/2000	912345678	ruif@gmail.com
										João	André	Rua da base de dados	12/5/2001	919876543	joaoa@gmail.com

Tabela 14 - Forma não normalizada – Registo Funcionário

1. Primeira Forma Normal (1FN)

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
E1	Estufa1	Porto	200	30	10	S1	Secção das frutas	50	F200	Rui	Fernandes	Rua da base de dados	2/3/2000	912345678	ruif@gmail.com
E1	Estufa1	Porto	200	30	10	S1	Secção das frutas	50	F201	João	André	Rua da base de dados	12/5/2001	919876543	joaoa@gmail.com

Tabela 15 - 1ª forma normalizada - Registo de funcionário

Primary key – A,G,J

Identificar dependências parciais:

A – B, C, D, E, F

G – H, I

J – K, L, M, N, O, P

2. Segunda Forma Normal (2FN)

Informação Estufa -

A	B	C	D	E	F
---	---	---	---	---	---

Informação Secção -

G	H	I
---	---	---

Informação funcionário -

J	K	L	M	N	O	P
---	---	---	---	---	---	---

Identificar dependências transitivas:

Não foram encontradas dependências transitivas.

3ª Forma Normal (3FN)

A terceira forma normal (3FN) elimina dependências transitivas. Nenhum atributo não chave deve depender de outro atributo não chave. Aqui, cada atributo não chave deve depender apenas da chave primária.

Com isso, podemos confirmar que as tabelas já estão em 3NF.

Relatório da plantação		
Plantação		
Id_Plantio: P101		Data início plantio:1/3/2024
Quantidade sementes:500	Colhido: Não	Em uso: sim
Produto plantado		
Id_Produto: P10		NomeP: tomate
DescriçãoP: Variedade tomate cereja		Duração: 90 dias
Id_Tipo: T3	NomeT: fruta	
Consumíveis utilizados		
Id_PlantioConsumivel: PC1	Id_Consumivel: C1	NomeC: Fertilizante A
Quantidade: 20	DescriçãoC: fertilizante orgânico	
Id_PlantioConsumivel: PC2	Id_Consumivel: C2	NomeC: Fertilizante B
Quantidade: 32	DescriçãoC: fertilizante não orgânico	

Tabela 16 - Documento do relatório da plantação

Passos de Normalização

Id_Plantio – A

Data início plantio– B

Quantidade sementes - C

Colhido – D

Em uso – E

Id_Produto – F

NomeP – G

DescriçãoP – H

Duração – I

Id_Tipo – J

NomeT – K

Id_PlantioConsumivel – L

Quantidade – M

Id_Consumivel – N

NomeC – O

DescriçãoC – P

Forma Não Normalizada

Forma inicial do documento onde há redundância e possíveis grupos repetidos. Vamos normalizar isso passo a passo.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
P101	1/3/2024	500	Não	Sim	P10	tomate	Variedade tomate cereja	90	T3	fruta	PC1	20	C1	Fertilizante A	fertilizante orgânico
											PC2	32	C2	Fertilizante B	fertilizante não orgânico

Tabela 17 - Forma não normalizada - relatório plantio

1ª Forma Normal (1FN)

A primeira forma normal (1 FN) requer que cada campo contenha apenas um valor e que cada registro seja único. Isso elimina grupos repetidos e garante a atomicidade dos dados.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
P101	1/3/2024	500	Não	Sim	P10	tomate	Variedade tomate cereja	90	T3	fruta	PC1	20	C1	Fertilizante A	fertilizante orgânico
P101	1/3/2024	500	Não	Sim	P10	tomate	Variedade tomate cereja	90	T3	fruta	PC2	32	C2	Fertilizante B	fertilizante não orgânico

Tabela 18 - 1ª forma normalizada - relatório plantio

Primary key – A, F, J, L

Identificar dependências parciais:

A – B, C, D, E

F – G, H, I, J, K

L – M, N, O, P

2ª Forma Normal (2FN)

Plantação -

A	B	C	D	E
---	---	---	---	---

Produto plantado -

F	G	H	I	J	K
---	---	---	---	---	---

Consumíveis utilizados -

L	M	N	O	P
---	---	---	---	---

Identificar dependências transitivas:

J -> K

N -> O, P

3ª Forma Normal (3FN)

Tipo -

J	K
---	---

PK - J

Consumível referenciado (consumível)

N	O	P
---	---	---

PK - N

Plantação -

A	B	C	D	E
---	---	---	---	---

PK - A

Produto plantado -

F	G	H	I	J
---	---	---	---	---

PK - F

FK - J

Consumíveis utilizados (PlantioConsumível) -

L	M	N
---	---	---

PK - L

FK - N

Relatório das colheitas de uma plantação		
Informação sobre a Plantação		
Id_Plantio: P101	Data início plantio:1/3/2024	
Quantidade sementes:500	Colhido: Não	Em uso: sim
Colheitas		
Id_Colheita: C202	Data colheita: 1/7/2024	Quantidade colhida: 450
Observações: Primeira colheita		
Id_Colheita: C203	Data colheita: 21/7/2024	Quantidade colhida: 180
Observações: Segunda colheita		
Produto colhido		
Id_Produto: P10	NomeP: tomate	
DescriçãoP: Variedade tomate cereja	Duração: 90 dias	
Id_Tipo: T3 NomeT: fruta		

Tabela 19 - Documento relatório colheitas de uma plantação

Passos de Normalização

Id_Plantio – A

Data início plantio – B

Quantidade sementes – C

Colhido – D

Em uso – E

Id_Colheita – F

Data colheita– G

Quantidade colhida - H

Observações – I

Id_Produto – J

NomeP – K

DescriçãoP – L

Duração – M

Id_Tipo – N

NomeT – O

Forma Não Normalizada

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
P101	1/3/2024	500	Não	Sim	C202	1/7/2024	450	Primeira colheita	P10	tomate	Variedade tomate cereja	90	T3	fruta
					C203	21/7/2024	180	Segunda colheita						

Tabela 20 - Forma não normalizada - relatório colheitas de uma plantação

1ª Forma Normal (1FN)

A primeira forma normal (1FN) requer que cada campo contenha apenas um valor e que cada registo seja único. Isso elimina grupos repetidos e garante a atomicidade dos dados.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
P101	1/3/2024	500	Não	Sim	C202	1/7/2024	450	Primeira colheita	P10	tomate	Variedade tomate cereja	90	T3	fruta
P101	1/3/2024	500	Não	Sim	C203	21/7/2024	180	Segunda colheita	P10	tomate	Variedade tomate cereja	90	T3	fruta

Tabela 21 - 1ª forma normalizada - Relatório colheitas de uma plantação

Primary key – A, F, J

Identificar dependências parciais:

A – B, C, D, E

F – G, H, I

J – K, L, M, N, O

2ª Forma Normal (2FN)

Informação sobre a Plantação –

A	B	C	D	E
---	---	---	---	---

Colheitas -

F	G	H	I
---	---	---	---

Produto colhido –

J	K	L	M	N	O
---	---	---	---	---	---

Identificar dependências transitivas:

N -> O

.

3ª Forma Normal (3FN)

Tipo -

N	O
---	---

PK – N

Informação sobre a Plantação -

A	B	C	D	E
---	---	---	---	---

PK – A

Colheitas -

F	G	H	I
---	---	---	---

PK – F

Produto colhido –

J	K	L	M	N
---	---	---	---	---

PK – J

FK – N

Modelo de dados lógico no final da normalização

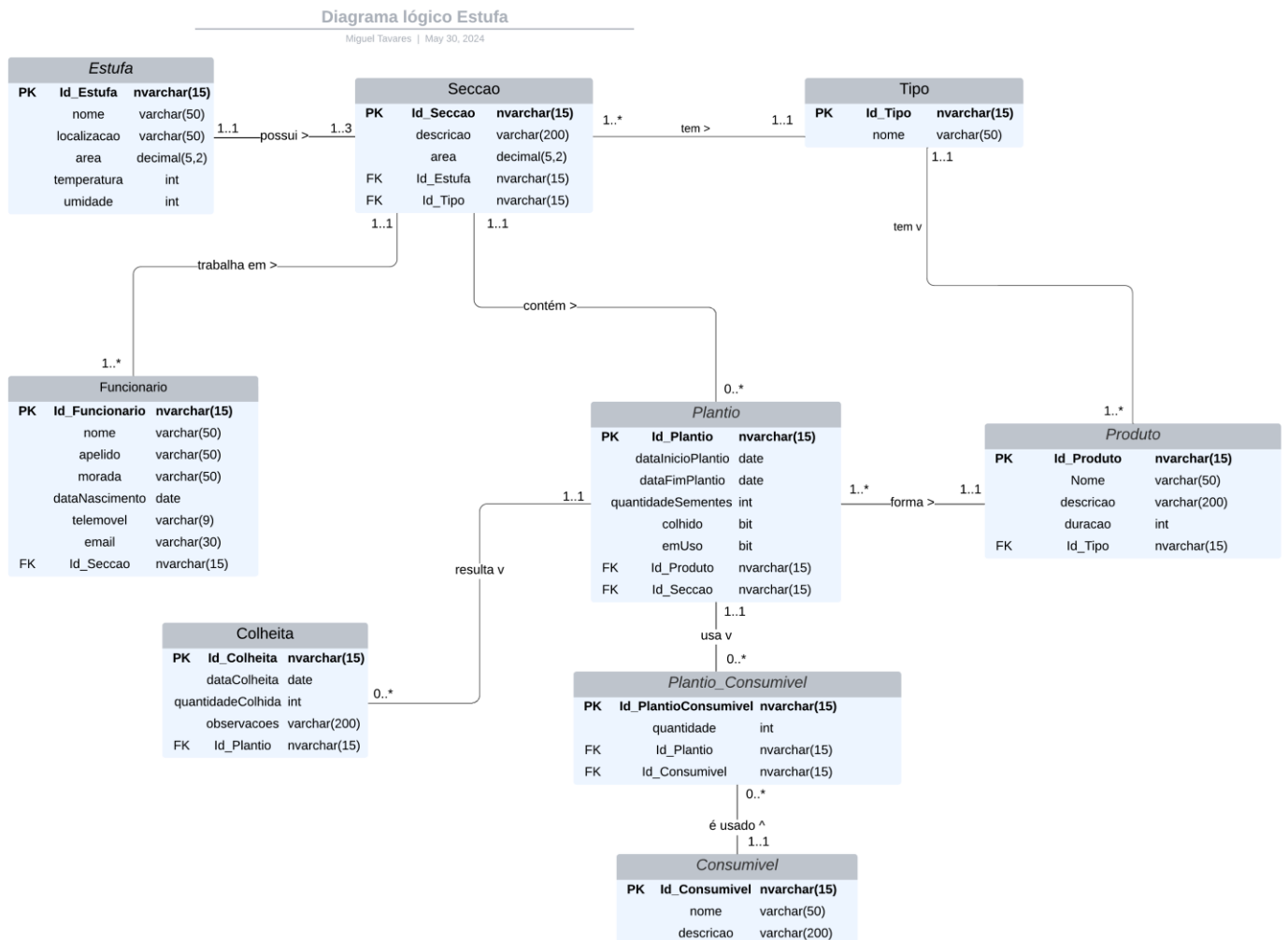


Figura 4 - Modelo logico de dados lógico depois da normalização

3.4 Validar novamente o modelo com transações

Após a implementação das relações normalizadas no modelo de dados, é crucial validar novamente o modelo com transações práticas para garantir a integridade e eficiência do sistema. Esta etapa envolve a execução de diversas transações que simulam operações reais para verificar a consistência dos dados e o correto funcionamento das relações estabelecidas.

Em conclusão, a validação do modelo com transações práticas confirmou que a estrutura de dados desenvolvida é sólida, eficiente e adequada para suportar as operações diárias de gestão de estufas, atendendo aos requisitos especificados no projeto.

3.5 Identificar Restrições de Integridade

Tabela: Estufa

Entidade	Atributo	Domínio	Null	Multivalores
Estufa	Id_Estufa	nvarchar(15)	Não	Não
	Nome	varchar(50)	Não	Não
	Localização	varchar(50)	Não	Não
	area	Decimal(5,2)	Não	Não
	Temperature	int	Não	Não
	Umidade	int	Não	Não

Tabela 22 - Entidade Estufa e as suas restrições de integridade

Tabela: Secção

Entidade	Atributo	Domínio	Null	Multivalores
Secção	Id_Seccao	nvarchar(15)	Não	Não
	Descrição	Varchar(200)	Não	Não
	area	decimal(5,2)	Não	Não
	Id_estufa	nvarchar(15)	Não	Não
	Id_Tipo	nvarchar(15)	Não	Não

Tabela 23 - Entidade Secção e as suas restrições de integridade

Tabela: Tipo

Entidade	Atributo	Domínio	Null	Multivalores
Tipo	Id_Tipo	nvarchar(15)	Não	Não
	nome	varchar(50)	Não	Não

Tabela 24 - Entidade Tipo e as suas restrições de integridade

Tabela: Produto

Entidade	Atributo	Domínio	Null	Multivalores
Produto	Id_Produto	nvarchar(15)	Não	Não
	Nome	varchar(50)	Não	Não
	Decricao	varchar(50)	Não	Não
	duracao	numType	Não	Não
	Id_Tipo	nvarchar(15)	Não	Não

Tabela 25 - Entidade Produto e as suas restrições de integridade

Tabela: Plantio

Entidade	Atributo	Domínio	Null	Multivalores
Plantio	Id_Plantio	nvarchar(15)	Não	Não
	dataInicioPlantio	date	Não	Não
	dataFimPlantio	date	Sim	Não
	quantidadeSementes	int	Não	Não
	Colhido	bit	Não	Não
	emUso	bit	Não	Não
	Id_Produto	nvarchar(15)	Não	Não
	Id_Seccao	nvarchar(15)	Não	Não

Tabela 26 - Entidade Plantio e as suas restrições de integridade

Tabela: Colheita

Entidade	Atributo	Domínio	Null	Multivalores
Colheita	Id_Colheita	nvarchar(15)	Não	Não
	dataColheita	date	Não	Não
	quantidadeColhida	int	Não	Não
	Observações	Varchar(200)	Não	Não
	Id_plantio	nvarchar(15)	Não	Não

Tabela 27 - Entidade Colheita e as suas restrições de integridade

Tabela: Consumivel

Entidade	Atributo	Domínio	Null	Multivalores
Consumivel	Id_Consumivel	nvarchar(15)	Não	Não
	Nome	varchar(50)	Não	Não
	descricao	varchar(200)	Não	Não

Tabela 28 - Entidade Consumivel e as suas restrições de integridade

Tabela: Plantio_Consumivel

Entidade	Atributo	Domínio	Null	Multivalores
Plantio_Consumivel	Id_Plantio_Consumivel	nvarchar(15)	Não	Não
	quantidade	int	Não	Não
	Id_Plantio	nvarchar(15)	Não	Não
	Id_Consumivel	nvarchar(15)	Não	Não

Tabela 29 - Entidade Plantio_Consumivel e as suas restrições de integridade

Tabela: Funcionario

Entidade	Atributo	Domínio	Null	Multivalores
Funcionario	Id_Funcionario	nvarchar(15)	Não	Não
	Nome	varchar(50)	Não	Não
	apelido	varchar(50)	Não	Não
	morada	varchar(50)	Não	Não
	dataNascimento	date	Não	Não
	telemovel	varchar(9)	Não	Não
	email	varchar(30)	Não	Não
	Id_Seccao	nvarchar(15)	Não	Não

Tabela 30 - Entidade Funcionario e as suas restrições de integridade

3.6 Regras de negócio

- Uma estufa pode conter no máximo 3 secções.
- Cada secção pode conter no máximo 20 plantios em uso.
- Cada secção só pode conter no máximo 10 produtos diferentes com plantações ativas.
- O tipo do produto a ser plantado, tem de ser igual ao tipo da secção.
- A data de fim de plantio é calculada ao somar 15 dias depois da mesma ser dada como colhida.
- Um plantio não tem limite de colheitas.
- A partir do momento que um plantio é dado como colhido, não podemos realizar mais colheitas relativas a essa plantação.
- Uma plantação pode usar vários consumíveis.
- Um funcionário trabalha apenas numa secção de uma estufa, ou seja, o mesmo não pode trabalhar em várias secções.
- Uma secção pode ter vários funcionários.

3.7 Validação do modelo com o utilizador

Tendo por base todos os pontos anteriores é possível dizer que o modelo é válido e cumpre todas as necessidades propostas logo é possível proceder para as próximas etapas.

3.8 Verificação do provável crescimento futuro

É fundamental garantir que o modelo lógico desenvolvido até o momento seja capaz de suportar as transações atuais e possíveis adições futuras com facilidade. Isso assegura que a base de dados atenderá às necessidades das estufas por um longo período. Considerando que, muitas vezes, o próprio utilizador não sabe exatamente quais serão as futuras necessidades, não é necessário alterar o modelo lógico desenvolvido até o momento, a menos que o mesmo solicite uma mudança específica.

4. Construção do modelo físico

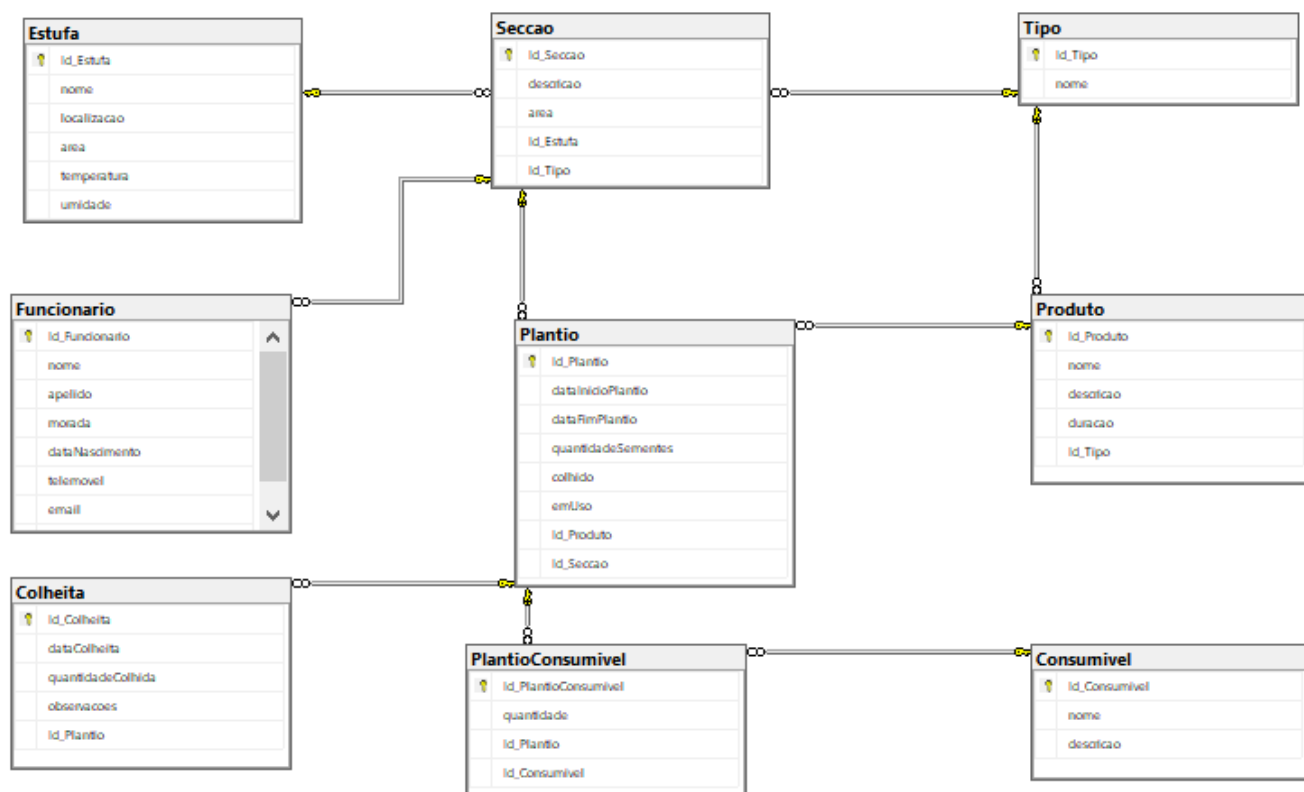


Figura 5 - Diagrama ER do modelo físico noSQL

4.1 Criação das tabelas no SQL Server Management Studio

- Tabela Estufa

```
USE [Projeto_BD]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Estufa](
    [Id_Estufa] [nvarchar](15) NOT NULL,
    [nome] [varchar](50) NOT NULL,
    [localizacao] [varchar](50) NOT NULL,
    [area] [decimal](5, 2) NOT NULL,
    [temperatura] [int] NOT NULL,
    [umidade] [int] NOT NULL,
    CONSTRAINT [PK_Estufa] PRIMARY KEY CLUSTERED
(
    [Id_Estufa] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY]
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Estufa] WITH CHECK ADD CONSTRAINT [Range Temperatura] CHECK
(((temperatura)>=(0) AND [temperatura]<=(100)))
GO
ALTER TABLE [dbo].[Estufa] CHECK CONSTRAINT [Range Temperatura]
GO
ALTER TABLE [dbo].[Estufa] WITH CHECK ADD CONSTRAINT [Range Umidade] CHECK
(((umidade)>=(0) AND [umidade]<=(100)))
GO
ALTER TABLE [dbo].[Estufa] CHECK CONSTRAINT [Range Umidade]
GO
ALTER TABLE [dbo].[Estufa] WITH CHECK ADD CONSTRAINT [Valor Area] CHECK
(((area)>(0)))
GO
ALTER TABLE [dbo].[Estufa] CHECK CONSTRAINT [Valor Area]
GO
```

Figura 6 - Create Table Estufa

- **Tabela Secção**

```
USE [Projeto_BD]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Seccao](
    [Id_Seccao] [nvarchar](15) NOT NULL,
    [descricao] [varchar](200) NOT NULL,
    [area] [decimal](5, 2) NOT NULL,
    [Id_Estufa] [nvarchar](15) NOT NULL,
    [Id_Tipo] [nvarchar](15) NOT NULL,
    CONSTRAINT [PK_Seccao] PRIMARY KEY CLUSTERED
(
    [Id_Seccao] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY]
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Seccao] WITH CHECK ADD CONSTRAINT [FK_Seccao_Estufa] FOREIGN
KEY([Id_Estufa])
REFERENCES [dbo].[Estufa] ([Id_Estufa])
GO
ALTER TABLE [dbo].[Seccao] CHECK CONSTRAINT [FK_Seccao_Estufa]
GO
ALTER TABLE [dbo].[Seccao] WITH CHECK ADD CONSTRAINT [FK_Seccao_Tipo] FOREIGN
KEY([Id_Tipo])
REFERENCES [dbo].[Tipo] ([Id_Tipo])
GO
ALTER TABLE [dbo].[Seccao] CHECK CONSTRAINT [FK_Seccao_Tipo]
GO
ALTER TABLE [dbo].[Seccao] WITH CHECK ADD CONSTRAINT [Area Value] CHECK
(((area)>(0)))
GO
ALTER TABLE [dbo].[Seccao] CHECK CONSTRAINT [Area Value]
GO
```

Figura 7 - Create Table Secção

- **Tabela Tipo**

```
USE [Projeto_BD]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Tipo](
    [Id_Tipo] [nvarchar](15) NOT NULL,
    [nome] [varchar](50) NOT NULL,
    CONSTRAINT [PK_Tipo] PRIMARY KEY CLUSTERED
(
    [Id_Tipo] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY]
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Tipo] WITH CHECK ADD CONSTRAINT [chk_Tipo_Id_Tipo] CHECK
((([Id_Tipo]='FR' OR [Id_Tipo]='PL' OR [Id_Tipo]='LE'))
GO
ALTER TABLE [dbo].[Tipo] CHECK CONSTRAINT [chk_Tipo_Id_Tipo]
GO
```

Figura 8 - Create Table Tipo

- **Tabela Produto**

```
USE [Projeto_BD]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Produto](
    [Id_Produto] [nvarchar](15) NOT NULL,
    [nome] [varchar](50) NOT NULL,
    [descricao] [varchar](200) NOT NULL,
    [duracao] [int] NOT NULL,
    [Id_Tipo] [nvarchar](15) NOT NULL,
    CONSTRAINT [PK_Produto] PRIMARY KEY CLUSTERED
(
    [Id_Produto] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY]
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Produto] WITH CHECK ADD CONSTRAINT [FK_Produto_Tipo] FOREIGN
KEY([Id_Tipo])
REFERENCES [dbo].[Tipo] ([Id_Tipo])
GO
ALTER TABLE [dbo].[Produto] CHECK CONSTRAINT [FK_Produto_Tipo]
GO
ALTER TABLE [dbo].[Produto] WITH CHECK ADD CONSTRAINT [Value Duracao] CHECK
((([duracao]>(0)))
GO
ALTER TABLE [dbo].[Produto] CHECK CONSTRAINT [Value Duracao]
GO
```

Figura 9 - Create Table Produto

- **Tabela Plantio**

```

USE [Projeto_BD]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Plantio](
    [Id_Plantio] [nvarchar](15) NOT NULL,
    [dataInicioPlantio] [date] NOT NULL,
    [dataFimPlantio] [date] NULL,
    [quantidadeSementes] [int] NOT NULL,
    [colhido] [bit] NOT NULL,
    [emUso] [bit] NOT NULL,
    [Id_Produto] [nvarchar](15) NOT NULL,
    [Id_Seccao] [nvarchar](15) NOT NULL,
    CONSTRAINT [PK_Plantio] PRIMARY KEY CLUSTERED
(
    [Id_Plantio] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY]
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Plantio] ADD CONSTRAINT [DF_Plantio_colhido] DEFAULT ((0))
FOR [colhido]
GO
ALTER TABLE [dbo].[Plantio] ADD CONSTRAINT [DF_Plantio_emUso] DEFAULT ((1)) FOR
[emUso]
GO
ALTER TABLE [dbo].[Plantio] WITH CHECK ADD CONSTRAINT [FK_Plantio_Produto]
FOREIGN KEY([Id_Produto])
REFERENCES [dbo].[Produto] ([Id_Produto])
GO
ALTER TABLE [dbo].[Plantio] CHECK CONSTRAINT [FK_Plantio_Produto]
GO
ALTER TABLE [dbo].[Plantio] WITH CHECK ADD CONSTRAINT [FK_Plantio_Seccao]
FOREIGN KEY([Id_Seccao])
REFERENCES [dbo].[Seccao] ([Id_Seccao])
GO
ALTER TABLE [dbo].[Plantio] CHECK CONSTRAINT [FK_Plantio_Seccao]
GO
ALTER TABLE [dbo].[Plantio] WITH CHECK ADD CONSTRAINT [Value
QuantidadeSementes] CHECK (([quantidadeSementes]>(0)))
GO
ALTER TABLE [dbo].[Plantio] CHECK CONSTRAINT [Value QuantidadeSementes]
GO

```

Figura 10 - Create Table Plantio

- **Tabela PlantioConsumivel**

```

USE [Projeto_BD]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[PlantioConsumivel](
    [Id_PlantioConsumivel] [nvarchar](15) NOT NULL,
    [quantidade] [int] NOT NULL,
    [Id_Plantio] [nvarchar](15) NOT NULL,
    [Id_Consumivel] [nvarchar](15) NOT NULL,
    CONSTRAINT [PK_PlantioConsumivel] PRIMARY KEY CLUSTERED
(
    [Id_PlantioConsumivel] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY]
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[PlantioConsumivel] WITH CHECK ADD CONSTRAINT
[FK_PlantioConsumivel_Consumivel] FOREIGN KEY([Id_Consumivel])
REFERENCES [dbo].[Consumivel] ([Id_Consumivel])
GO
ALTER TABLE [dbo].[PlantioConsumivel] CHECK CONSTRAINT
[FK_PlantioConsumivel_Consumivel]
GO
ALTER TABLE [dbo].[PlantioConsumivel] WITH CHECK ADD CONSTRAINT
[FK_PlantioConsumivel_Plantio] FOREIGN KEY([Id_Plantio])
REFERENCES [dbo].[Plantio] ([Id_Plantio])
GO
ALTER TABLE [dbo].[PlantioConsumivel] CHECK CONSTRAINT
[FK_PlantioConsumivel_Plantio]
GO
ALTER TABLE [dbo].[PlantioConsumivel] WITH CHECK ADD CONSTRAINT [Value
Quantidade] CHECK (([quantidade]>(0)))
GO
ALTER TABLE [dbo].[PlantioConsumivel] CHECK CONSTRAINT [Value Quantidade]
GO

```

Figura 11 - Create Table Plantio Consumivel

- **Tabela Consumivel**

```
USE [Projeto_BD]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Consumivel](
    [Id_Consumivel] [nvarchar](15) NOT NULL,
    [nome] [varchar](50) NOT NULL,
    [descricao] [varchar](200) NOT NULL,
    CONSTRAINT [PK_Consumivel] PRIMARY KEY CLUSTERED
(
    [Id_Consumivel] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY]
) ON [PRIMARY]
GO
```

Figura 12 - Create Table Consumivel

- **Tabela Colheita**

```
USE [Projeto_BD]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Colheita](
    [Id_Colheita] [nvarchar](15) NOT NULL,
    [dataColheita] [date] NOT NULL,
    [quantidadeColhida] [int] NOT NULL,
    [observacoes] [varchar](200) NOT NULL,
    [Id_Plantio] [nvarchar](15) NOT NULL,
    CONSTRAINT [PK_Colheita] PRIMARY KEY CLUSTERED
(
    [Id_Colheita] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY]
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Colheita] WITH CHECK ADD CONSTRAINT [FK_Colheita_Plantio]
FOREIGN KEY([Id_Plantio])
REFERENCES [dbo].[Plantio] ([Id_Plantio])
GO
ALTER TABLE [dbo].[Colheita] CHECK CONSTRAINT [FK_Colheita_Plantio]
GO
ALTER TABLE [dbo].[Colheita] WITH CHECK ADD CONSTRAINT [Value
QuantidadeColhida] CHECK (([quantidadeColhida]>(0)))
GO
ALTER TABLE [dbo].[Colheita] CHECK CONSTRAINT [Value QuantidadeColhida]
GO
```

Figura 13 - Create Table Colheita

- **Tabela Funcionario**

```

USE [Projeto_BD]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Funcionario](
    [Id_Funcionario] [nvarchar](15) NOT NULL,
    [nome] [varchar](50) NOT NULL,
    [apelido] [varchar](50) NOT NULL,
    [morada] [varchar](50) NOT NULL,
    [dataNascimento] [date] NOT NULL,
    [telemovel] [varchar](9) NOT NULL,
    [email] [varchar](30) NOT NULL,
    [Id_Seccao] [nvarchar](15) NOT NULL,
    CONSTRAINT [PK_Funcionario] PRIMARY KEY CLUSTERED
(
    [Id_Funcionario] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY]
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Funcionario] WITH CHECK ADD CONSTRAINT
[FK_Funcionario_Seccao] FOREIGN KEY([Id_Seccao])
REFERENCES [dbo].[Seccao] ([Id_Seccao])
GO
ALTER TABLE [dbo].[Funcionario] CHECK CONSTRAINT [FK_Funcionario_Seccao]
GO

```

Figura 14 - Create Table Funcionario

4.2 Criação das procedures de CUD (Create, Update, Delete) no SQL Server Management Studio

Seguem-se alguns exemplos de procedures

- **Atualizar colheita**

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[AtualizarColheita]
    @Id_Colheita NVARCHAR(15),
    @dataColheita DATE,
    @quantidadeColhida INT,
    @observacoes NVARCHAR(200),
    @Id_Plantio NVARCHAR(15)
AS
BEGIN
    BEGIN TRY
        BEGIN TRANSACTION

            UPDATE Colheita
            SET dataColheita = @dataColheita,
                quantidadeColhida = @quantidadeColhida,
                observacoes = @observacoes,
                Id_Plantio = @Id_Plantio
            WHERE Id_Colheita = @Id_Colheita

        COMMIT TRANSACTION
    END TRY
    BEGIN CATCH
        ROLLBACK TRANSACTION
        DECLARE @ErrorMessage NVARCHAR(4000) = ERROR_MESSAGE();
        RAISERROR('Erro ao atualizar colheita: %s', 16, 1, @ErrorMessage)
    END CATCH
END
GO
```

Figura 15 - Procedure Atualizar Colheita

- **Eliminar colheita**

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[EliminarColheita]
    @Id_Colheita NVARCHAR(15)
AS
BEGIN
    BEGIN TRY
        BEGIN TRANSACTION

        DELETE FROM Colheita
        WHERE Id_Colheita = @Id_Colheita

        COMMIT TRANSACTION
    END TRY
    BEGIN CATCH
        ROLLBACK TRANSACTION
        DECLARE @ErrorMessage NVARCHAR(4000) = ERROR_MESSAGE();
        RAISERROR('Erro ao eliminar colheita: %s', 16, 1, @ErrorMessage)
    END CATCH
END
GO
```

Figura 16 - Procedure Eliminar Colheita

- **Inserir colheita**

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[InserirColheita]
    @Id_Colheita NVARCHAR(15),
    @dataColheita DATE,
    @quantidadeColhida INT,
    @observacoes NVARCHAR(200),
    @Id_Plantio NVARCHAR(15)
AS
BEGIN
    BEGIN TRY
        BEGIN TRANSACTION

        INSERT INTO Colheita (Id_Colheita, dataColheita, quantidadeColhida,
observacoes, Id_Plantio)
        VALUES (@Id_Colheita, @dataColheita, @quantidadeColhida, @observacoes,
@Id_Plantio)

        COMMIT TRANSACTION
    END TRY
    BEGIN CATCH
        ROLLBACK TRANSACTION
        DECLARE @ErrorMessage NVARCHAR(4000) = ERROR_MESSAGE();
        RAISERROR('Erro ao inserir colheita: %s', 16, 1, @ErrorMessage)
    END CATCH
END
GO
```

Figura 17 - Procedure Inserir Colheita

- **Atualizar Estufa**

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[AtualizarEstufa]
    @Id_Estufa NVARCHAR(15),
    @nome NVARCHAR(50),
    @localizacao NVARCHAR(100),
    @area DECIMAL(5, 2),
    @temperatura DECIMAL(5, 2),
    @umidade DECIMAL(5, 2)
AS
BEGIN
    BEGIN TRY
        BEGIN TRANSACTION

            UPDATE Estufa
            SET nome = @nome,
                localizacao = @localizacao,
                area = @area,
                temperatura = @temperatura,
                umidade = @umidade
            WHERE Id_Estufa = @Id_Estufa

        COMMIT TRANSACTION
    END TRY
    BEGIN CATCH
        ROLLBACK TRANSACTION
        DECLARE @ErrorMessage NVARCHAR(4000) = ERROR_MESSAGE();
        RAISERROR('Erro ao atualizar estufa: %s', 16, 1, @ErrorMessage)
    END CATCH
END
GO

```

Figura 18 - Procedure Atualizar Estufa

- **Eliminar Estufa**

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[EliminarEstufa]
    @Id_Estufa NVARCHAR(15)
AS
BEGIN
    BEGIN TRY
        BEGIN TRANSACTION

            DELETE FROM Estufa
            WHERE Id_Estufa = @Id_Estufa

        COMMIT TRANSACTION
    END TRY
    BEGIN CATCH
        ROLLBACK TRANSACTION
        DECLARE @ErrorMessage NVARCHAR(4000) = ERROR_MESSAGE();
        RAISERROR('Erro ao eliminar estufa: %s', 16, 1, @ErrorMessage)
    END CATCH
END

```

Figura 19 - Procedure Eliminar Estufa

- **Inserir Estufa**

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[InserirEstufa]
    @Id_Estufa NVARCHAR(15),
    @nome NVARCHAR(50),
    @localizacao NVARCHAR(100),
    @area DECIMAL(5, 2),
    @temperatura DECIMAL(5, 2),
    @umidade DECIMAL(5, 2)
AS
BEGIN
    BEGIN TRY
        BEGIN TRANSACTION

            INSERT INTO Estufa (Id_Estufa, nome, localizacao, area, temperatura,
umidade)
VALUES (@Id_Estufa, @nome, @localizacao, @area, @temperatura, @umidade)

        COMMIT TRANSACTION
    END TRY
    BEGIN CATCH
        ROLLBACK TRANSACTION
        DECLARE @ErrorMessage NVARCHAR(4000) = ERROR_MESSAGE();
        RAISERROR('Erro ao inserir estufa: %s', 16, 1, @ErrorMessage)
    END CATCH
END
GO

```

Figura 20 - Procedure Inserir Estufa

- **Atualizar Plantio**

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[AtualizarPlantio]
    @Id_Plantio NVARCHAR(15),
    @dataInicioPlantio DATE,
    @dataFimPlantio DATE,
    @quantidadeSementes INT,
    @colhido BIT,
    @emUso BIT,
    @Id_Produto NVARCHAR(15),
    @Id_Seccao NVARCHAR(15)
AS
BEGIN
    BEGIN TRY
        BEGIN TRANSACTION

            UPDATE Plantio

```

```

SET dataInicioPlantio = @dataInicioPlantio,
    dataFimPlantio = @dataFimPlantio,
    quantidadeSementes = @quantidadeSementes,
    colhido = @colhido,
    emUso = @emUso,
    Id_Produto = @Id_Produto,
    Id_Seccao = @Id_Seccao
WHERE Id_Plantio = @Id_Plantio

COMMIT TRANSACTION
END TRY
BEGIN CATCH
    ROLLBACK TRANSACTION
    DECLARE @ErrorMessage NVARCHAR(4000) = ERROR_MESSAGE();
    RAISERROR('Erro ao atualizar plantio: %s', 16, 1, @ErrorMessage)
END CATCH
END
GO

```

Figura 21 - Procedure Atualizar Plantio

- **Eliminar Plantio**

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[EliminarPlantio]
    @Id_Plantio NVARCHAR(15)
AS
BEGIN
    BEGIN TRY
        BEGIN TRANSACTION

        DELETE FROM Plantio
        WHERE Id_Plantio = @Id_Plantio

        COMMIT TRANSACTION
    END TRY
    BEGIN CATCH
        ROLLBACK TRANSACTION
        DECLARE @ErrorMessage NVARCHAR(4000) = ERROR_MESSAGE();
        RAISERROR('Erro ao eliminar plantio: %s', 16, 1, @ErrorMessage)
    END CATCH
END
GO

```

Figura 22 - Procedure Eliminar Plantio

- **Inserir Plantio**

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[InserirPlantio]
    @Id_Plantio NVARCHAR(15),
    @dataInicioPlantio DATE,
    @dataFimPlantio DATE,
    @quantidadeSementes INT,
    @colhido BIT,
    @emUso BIT,
    @Id_Produto NVARCHAR(15),
    @Id_Seccao NVARCHAR(15)
AS
BEGIN
    BEGIN TRY
        BEGIN TRANSACTION

            INSERT INTO Plantio (Id_Plantio, dataInicioPlantio, dataFimPlantio,
quantidadeSementes, colhido, emUso, Id_Produto, Id_Seccao)
            VALUES (@Id_Plantio, @dataInicioPlantio, @dataFimPlantio,
@quantidadeSementes, @colhido, @emUso, @Id_Produto, @Id_Seccao)

        COMMIT TRANSACTION
    END TRY
    BEGIN CATCH
        ROLLBACK TRANSACTION
        DECLARE @ErrorMessage NVARCHAR(4000) = ERROR_MESSAGE();
        RAISERROR('Erro ao inserir plantio: %s', 16, 1, @ErrorMessage)
    END CATCH
END
GO

```

Figura 23 - Procedure Inserir Plantio

- **Reverter Plantio**

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[ReverterPlantio]
AS
BEGIN
    SET NOCOUNT ON;

    -- Reverte os campos Colhido e EmUso após 15 dias
    UPDATE Plantio
    SET Colhido = 1,
        EmUso = 0
    WHERE Colhido = 1 AND DATEDIFF(DAY, dataFimPlantio, GETDATE()) >= 15;
END
GO

```

Figura 24 - Procedure Reverter Plantio

4.3 Consultas propostas

1. Quais os produtos colhidos no mês anterior.

```
SELECT P.nome AS Produto, C.dataColheita, C.quantidadeColhida, S.descricao AS Seccao, E.nome AS Estufa
FROM   dbo.Colheita AS C INNER JOIN
       dbo.Plantio AS PL ON C.Id_Plantio = PL.Id_Plantio INNER JOIN
       dbo.Produto AS P ON PL.Id_Produto = P.Id_Produto INNER JOIN
       dbo.Seccao AS S ON PL.Id_Seccao = S.Id_Seccao INNER JOIN
       dbo.Estufa AS E ON S.Id_Estufa = E.Id_Estufa
WHERE  (C.dataColheita >= DATEADD(MONTH, - 1, DATEADD(MONTH, DATEDIFF(MONTH, 0, GETDATE()), 0))) AND (C.dataColheita < DATEADD(MONTH, DATEDIFF(MONTH, 0, GETDATE()), 0))
```

Figura 25 - Consulta 'Quais os produtos colhidos no mês anterior'

	Produto	dataColheita	quantidadeColhida	Seccao	Estufa
1	Maça	2024-05-01	12	Aqui Temos Frutas	Estufa1

Figura 26 - Resultado exemplo da consulta 'Quais os produtos colhidos no mês anterior'

2. Qual a quantidade semeada por trimestre por produto no último ano

```
SELECT p.nome AS Produto, YEAR(pl.dataInicioPlantio) AS Ano, DATEPART(QUARTER, pl.dataInicioPlantio) AS Trimestre, SUM(pl.quantidadeSementes) AS QuantidadeSemeada
FROM   dbo.Plantio AS pl INNER JOIN
       dbo.Produto AS p ON pl.Id_Produto = p.Id_Produto
WHERE  (pl.dataInicioPlantio >= DATEFROMPARTS(YEAR(GETDATE()) - 1, 1, 1))
GROUP BY p.nome, YEAR(pl.dataInicioPlantio), DATEPART(QUARTER, pl.dataInicioPlantio)
```

Figura 27 - Consulta 'Qual a quantidade semeada por trimestre por produto no último ano'

	Produto	Ano	Trimestre	QuantidadeSemeada
1	Maça	2023	1	463
2	Maça	2023	2	550

Figura 28 - Resultado exemplo 'Qual a quantidade semeada por trimestre por produto no último ano'

3. Lista dos produtos com maior quantidade colhida no ano de 2018

```
SELECT P.nome AS Produto, SUM(C.quantidadeColhida) AS Quantidade_Total_Colhida
FROM   dbo.Colheita AS C INNER JOIN
       dbo.Plantio AS PL ON C.Id_Plantio = PL.Id_Plantio INNER JOIN
       dbo.Produto AS P ON PL.Id_Produto = P.Id_Produto
WHERE  (DATEPART(YEAR, C.dataColheita) = 2018)
GROUP BY P.nome
```

Figura 29 - Consulta 'Lista dos produtos com maior quantidade colhida no ano de 2018'

	Produto	Quantidade_Total_Colhida
--	---------	--------------------------

Figura 30 - Resultado exemplo da consulta lista dos produtos com maior quantidade colhida no ano de 2018'

Outras consultas relevantes:

- **Vista geral da agricultura**

```
SELECT E.nome AS Nome_Estufa, E.localizacao AS Localizacao_Estufa, S.descricao AS Descricao_Seccao, S.area AS Area_Seccao, P.nome AS Nome_Produto, PL.dataInicioPlantio, PL.dataFimPlantio,
       PL.quantidadeSementes AS Quantidade_Plantada, PL.colhido AS Produto_Colhido, C.dataColheita, C.quantidadeColhida AS Quantidade_Colhida, CON.nome AS Nome_Consumivel,
       PC.quantidade AS Quantidade_Consumivel_Usada
FROM   dbo.Estufa AS E INNER JOIN
       dbo.Seccao AS S ON E.Id_Estufa = S.Id_Estufa INNER JOIN
       dbo.Plantio AS PL ON S.Id_Seccao = PL.Id_Seccao INNER JOIN
       dbo.Produto AS P ON PL.Id_Produto = P.Id_Produto LEFT OUTER JOIN
       dbo.Colheita AS C ON PL.Id_Plantio = C.Id_Plantio LEFT OUTER JOIN
       dbo.PlantioConsumivel AS PC ON PL.Id_Plantio = PC.Id_Plantio LEFT OUTER JOIN
       dbo.Consumivel AS CON ON PC.Id_Consumivel = CON.Id_Consumivel|
```

Figura 31 - Consulta 'vista geral da agricultura'

	Nome_Estufa	Localizacao_Estufa	Descricao_Seccao	Area_Seccao	Nome_Produto	dataInicioPlantio	dataFimPlantio	Quantidade_Plantada	Produto_Colhido	dataColheita	Quantidade_Colhida	Nome_Consumi
1	Estufa1	Porto	Aqui Temos Frutas	40.00	Maça	2018-02-01	2018-04-03	13	0	2018-04-04	13	agua
2	Estufa1	Porto	Aqui Temos Frutas	40.00	Maça	2018-02-01	2018-04-03	13	0	2018-04-04	13	adubo
3	Estufa1	Porto	Aqui Temos Frutas	40.00	Maça	2018-02-01	2018-04-03	13	0	2024-05-01	12	agua
4	Estufa1	Porto	Aqui Temos Frutas	40.00	Maça	2018-02-01	2018-04-03	13	0	2024-05-01	12	adubo

Figura 32 - Exemplo de resultado da consulta 'vista geral agricultura'

4.3 Triggers

Trigger para verificar a área da secção em relação à estufa

A área tem de ser maior que 0 e menor que a área da estufa

```
USE [Projeto_BD]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER TRIGGER [dbo].[trg_VerificarAreaSecao]
ON [dbo].[Seccao]
FOR INSERT, UPDATE
AS
BEGIN
    DECLARE @Id_Estufa INT
    DECLARE @Area_Secao DECIMAL(10, 2)
    DECLARE @IsValid BIT

    -- Obter os valores das inserções ou atualizações
    SELECT @Id_Estufa = i.Id_Estufa, @Area_Secao = i.Area
    FROM inserted i

    -- Chamar a função para verificar a área
    SET @IsValid = dbo.fn_VerificarAreaSecao(@Id_Estufa, @Area_Secao)

    -- Verificar se a área da secção é válida
    IF @IsValid = 0
    BEGIN
        RAISERROR('A área da secção não pode ser maior que a área da estufa.',
16, 1)
        ROLLBACK TRANSACTION
    END
END
```

Figura 33 - trigger 'verificar area da estufa'

Trigger limite de secções por estufa

Uma estufa só pode conter no máximo 3 secções

```
USE [Projeto_BD]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER TRIGGER [dbo].[trg_LimitarSecoesPorEstufa]
ON [dbo].[Seccao]
FOR INSERT, UPDATE
AS
BEGIN
    DECLARE @Id_Estufa NVARCHAR(15)
    DECLARE @Count INT

    -- Seleciona o Id_Estufa da inserção ou atualização
    SELECT @Id_Estufa = i.Id_Estufa
    FROM inserted i

    -- Conta o número de secções na estufa
    SELECT @Count = COUNT(*)
    FROM Seccao
    WHERE Id_Estufa = @Id_Estufa

    -- Se o número de secções for maior que 3, lança um erro e faz rollback
    IF @Count > 3
    BEGIN
        RAISERROR('Cada estufa pode ter no máximo 3 secções.', 16, 1)
        ROLLBACK TRANSACTION
    END
END
END
```

Figura 34 - Trigger 'limite secções estufa'

Trigger para verificar tipo de Produto / Secção

Este trigger é responsável por verificar se o tipo do produto a ser plantado em determinada secção é igual ao tipo da secção, ou seja, produto e secção têm de ser do mesmo tipo.

```
USE [Projeto_BD]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER TRIGGER [dbo].[trg_VerificarTipoProdutoSecao]
ON [dbo].[Plantio]
FOR INSERT, UPDATE
AS
BEGIN
    DECLARE @Id_Seccao NVARCHAR(15)
    DECLARE @Id_Produto NVARCHAR(15)
    DECLARE @Tipo_Secao NVARCHAR(50)
    DECLARE @Tipo_Produto NVARCHAR(50)

    -- Seleciona o Id_Seccao e Id_Produto da inserção ou atualização
    SELECT @Id_Seccao = i.Id_Seccao, @Id_Produto = i.Id_Produto
    FROM inserted i

    -- Obtém o tipo da secção
    SELECT @Tipo_Secao = s.Id_Tipo
    FROM Seccao s
    WHERE s.Id_Seccao = @Id_Seccao

    -- Obtém o tipo do produto
    SELECT @Tipo_Produto = p.Id_Tipo
    FROM Produto p
    WHERE p.Id_Produto = @Id_Produto

    -- Verifica se os tipos coincidem
    IF @Tipo_Secao != @Tipo_Produto
    BEGIN
        RAISERROR('O tipo do produto não coincide com o tipo da secção.', 16, 1)
        ROLLBACK TRANSACTION
        RETURN
    END
END
```

Figura 35 - Trigger Verificar Tipo

Trigger que limita a quantidade de produtos por Secção

Este trigger serve para limitar a quantidade de produtos plantados durante um intervalo de tempo numa secção

Só se pode plantar 10 produtos diferentes por secção ao mesmo tempo.

```
USE [Projeto_BD]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

ALTER TRIGGER [dbo].[trg_LimitarProdutosPorTipoPorSecao]
ON [dbo].[Plantio]
FOR INSERT, UPDATE
AS
BEGIN
    DECLARE @Id_Seccao NVARCHAR(15)
    DECLARE @Id_Produto NVARCHAR(15)
    DECLARE @Id_Tipo NVARCHAR(15)
    DECLARE @Count INT

    -- Seleciona o Id_Seccao e Id_Produto da inserção ou atualização
    SELECT @Id_Seccao = i.Id_Seccao, @Id_Produto = i.Id_Produto
    FROM inserted i

    -- Seleciona o Id_Tipo do produto inserido ou atualizado
    SELECT @Id_Tipo = p.Id_Tipo
    FROM Produto p
    WHERE p.Id_Produto = @Id_Produto

    -- Conta o número de produtos diferentes do mesmo tipo na secção
    SELECT @Count = COUNT(DISTINCT p.Id_Produto)
    FROM Plantio pl
    JOIN Produto p ON pl.Id_Produto = p.Id_Produto
    WHERE pl.Id_Seccao = @Id_Seccao AND p.Id_Tipo = @Id_Tipo

    -- Se o número de produtos diferentes do mesmo tipo for maior que 10, lança
    um erro e faz rollback
    IF @Count > 10
    BEGIN
        RAISERROR('Cada secção pode ter no máximo 10 produtos diferentes de um
        mesmo tipo.', 16, 1)
        ROLLBACK TRANSACTION
    END
END
```

Figura 36 - Trigger Limite Produto / Secção

Trigger que limita o número de plantações por secção

Este trigger é responsável por verificar o número de plantações numa secção, a mesma só pode ter no máximo 20 plantações

```
USE [Projeto_BD]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER TRIGGER [dbo].[trg_LimitarPlantioPorSecao]
ON [dbo].[Plantio]
FOR INSERT, UPDATE
AS
BEGIN
    DECLARE @Id_Seccao NVARCHAR(15)
    DECLARE @PlantioCount INT

    -- Seleciona o Id_Seccao da inserção ou atualização
    SELECT @Id_Seccao = i.Id_Seccao
    FROM inserted i

    -- Conta quantas plantações existem para essa secção
    SELECT @PlantioCount = COUNT(*)
    FROM Plantio
    WHERE Id_Seccao = @Id_Seccao

    -- Se o número de plantações já for 20 ou mais, impede a operação
    IF @PlantioCount >= 20
    BEGIN
        RAISERROR('Cada secção pode ter no máximo 20 plantações.', 16, 1)
        ROLLBACK TRANSACTION
        RETURN
    END
END
END
```

Figura 37 - Trigger Limite Plantações Por Secção

Trigger atualizar data fim de plantio

Este trigger é responsável por atualizar a data de fim de plantio

```
USE [Projeto_BD]
GO
/***** Object:  Trigger [dbo].[trg_AtualizarDataFimPlantio]    Script Date:
06/06/2024 00:01:23 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER TRIGGER [dbo].[trg_AtualizarDataFimPlantio]
ON [dbo].[Plantio]
FOR UPDATE
AS
BEGIN
    IF UPDATE(Colhido)
    BEGIN
        DECLARE @Id_Plantio INT

        -- Seleciona os valores da inserção ou atualização
        SELECT @Id_Plantio = i.Id_Plantio
        FROM inserted i
        WHERE i.Colhido = 1

        -- Atualiza a Data_Fim com a data atual
        UPDATE Plantio
        SET dataFimPlantio = GETDATE()
        WHERE Id_Plantio = @Id_Plantio
    END
END
```

Figura 38 - Trigger Atualizar Data Fim Plantio

4.3 SQL SERVER AGENT

Aqui usamos o SQL server agente para verificar se já se passaram 15 dias de repouso após a data de fim de plantio para poder replantar, este executa todos os dias o stored procedure ReverterPlantio à 00:00.

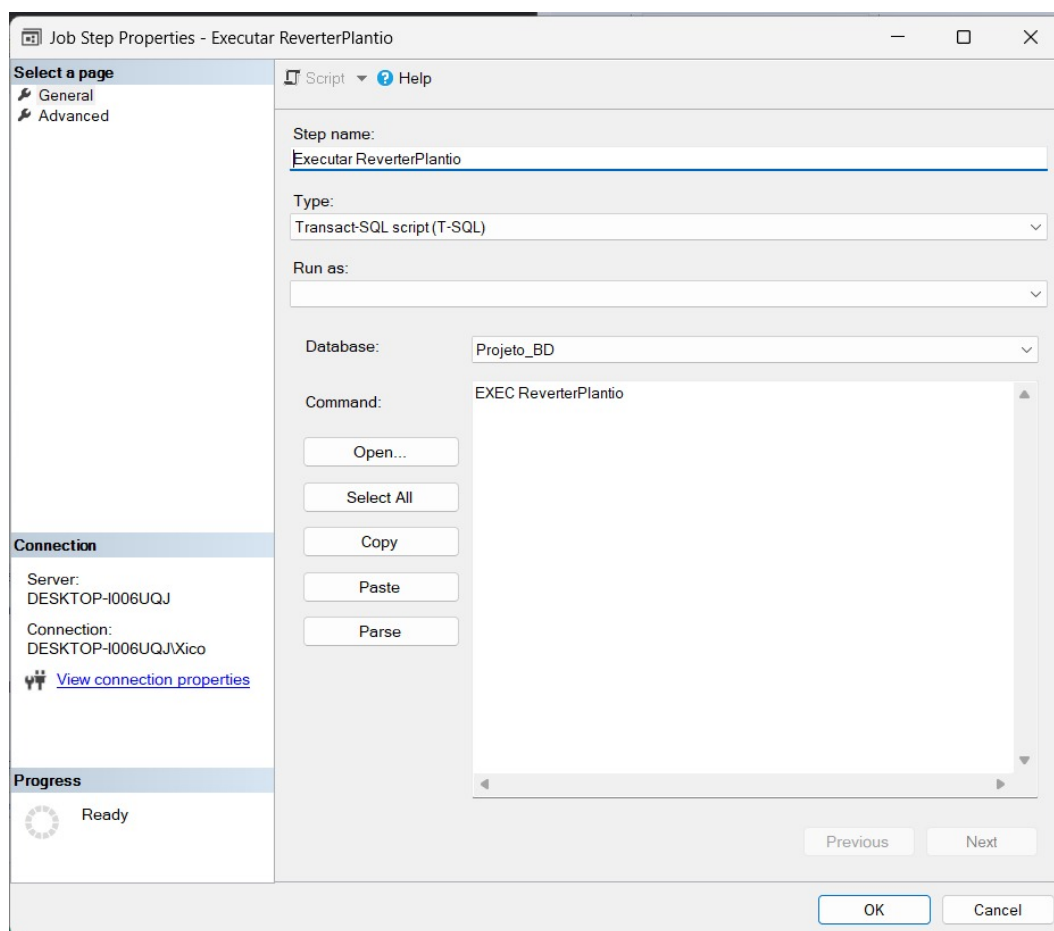


Figura 39-Step do SQL SERVER AGENT

Job Schedule Properties - Agendamento

Name: Jobs in Schedule

Schedule type: Recurring ☒ Enabled

One-time occurrence

Date: Time:

Frequency

Occurs: Daily

Recurs every: 1 day(s)

Daily frequency

☒ Occurs once at:

☐ Occurs every: 1 hour(s)

Starting at:

Ending at:

Duration

Start date: ☐ End date:

☒ No end date:

Summary

Description:

OK Cancel Help

Figura 40-Agendamento para execução do Job

5. Conclusões e Trabalho Futuro

Uma vez concluído o trabalho e aplicados todos os conceitos adquiridos durante o semestre na unidade curricular de Base de Dados foi possível obter uma visão mais prática de como é, e como deve funcionar, uma base de dados e todas as etapas da sua criação. Posto isto, também é possível verificar que existem alguns aspetos que devem ser melhorados de maneira que se possa satisfazer totalmente a base de dados em questão. Em conclusão, o trabalho foi concluído satisfatoriamente, abrangendo todos os pontos propostos.

Bibliografia

C. E. B. THOMAS M. CONNOLLY, DATABASE SYSTEMS -A Practical Approach to Design, Implementation, and Management, Pearson, 2015.

Referências WWW

[01] <https://moodle2.estg.ipp.pt/course/view.php?id=214>

Página do Moodle da Unidade Curricular onde encontramos os conteúdos sobre BD.

[02] <https://stackoverflow.com/>

[03] <https://www.youtube.com/playlist?list=PLTU76O6uRMfQixOnW0IAdDxwCM40M8rTI>

Playlist de vídeos disponibilizados pelo professor – Tutoriais Vídeos SQL Server 2017

[04]

https://www.youtube.com/watch?v=ZNObiptSMSI&list=PL08903FB7ACA1C2FB&ab_channel=kudvenkat

Playlist de vídeos disponibilizados pelo professor - Tutoriais Vídeos SQL Server - mais antigo mas mais completo

Lista de Siglas e Acrónimos

SGBD Sistema de Gestão de Base de Dados

BD Base de Dados

DBDL Database Designation Language (Linguagem de designação de base de dados)

PK Primary Key (Chave Primária)

FK Foreign key (Chave Estrangeira)

UNF Forma não normalizada

1FN 1ª forma normalizada

2FN 2ª forma normalizada

3FN 3ª forma normalizada