

23,0 / 25,0

UNIVERSIDADE FEDERAL DE OURO PRETO — DEPARTAMENTO DE COMPUTAÇÃO

BCC 321 - BANCO DE DADOS 1

Gerenciamento de Fazenda

Antônio Alex Gomes Rodrigues - 19.1.4173

Daniel Monteiro Valério - 19.1.4035

Gabriel Fernandes Niquini 19.1.4113

Fábio Henrique Soares - 19.1.4008

Rômulo de Oliveira Carneiro - 19.1.4107

OURO PRETO

2022

1. Mini-mundo

**1,0 / 1,0
sem alteração**

O problema proposto é baseado no gerenciamento de fazendas de um produtor, onde cada produtor deverá se cadastrar no sistema com username (Único), primeiro nome, último nome, email e senha.

Ao estar logado ele precisará criar uma fazenda para liberar a funcionalidade dos galpões. Uma fazenda terá um id (Único), nome, país e estado e poderá conter vários galpões que por sua vez, terão um id (Único), nome, data de construção, tamanho, capacidade e cidade mais próxima.

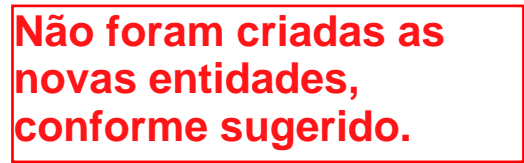
Após os galpões criados, o produtor irá entrar com os dados de seus Flocks, onde cada um deles terá um id (Único), data de nascimento, número de aves alocadas, tipo, strain, sistema, fonte de água.

A documentação referente ao flock terá um id (Único), um nome, categoria e o arquivo PDF relacionado, essa documentação é opcional. Se necessário poderá se requisitar mais aves ou vender o excedente, cada pedido também terá um id (Único), a data, quantidade, preço para requisição ou venda, caso requisição deverá ter também preço de frete.

Para acompanhar o progresso de seus Flocks, o produtor terá que entrar com dados periodicamente sobre a performance e produção das aves. Sendo as métricas de performance um id (Único), a data de input, data inicial e final do período referente aos dados, número de aves mortas, quantidade de ração e água consumida, peso das aves e ração utilizada. Já as métricas de produção são um id (Único), a data de input dos dados, data inicial e final do período referente, quantidade de ovos jumbo, extra grande, grande, médio e pequeno, bem como o preço desses mesmos ovos.

As rações usadas para atualizar os dados de performance periodicamente devem fazer parte do estoque do silo no galpão, que por sua vez guardam um id (Único), datação do começo da alimentação, número do silo, quantidade, custo por tonelada, custo de transporte, toneladas restantes e deve estar associado a uma dieta. A dieta por sua vez possui um id (Único) nome, empresa vendedora, tipo de ração e granulometria e caso seja do tipo completa deve-se ter também os campos de quantidade de proteína e subsídio da alimentação especificados.

2,8 / 3,0



3. Dicionário de Entidades

Entidade	Semântica	Lista de Atributos
Produtor	O produtor é o usuário do sistema	Username, Nome, Email e Senha
Fazenda	As Fazendas possuem a localização das fazendas pertencentes a um respectivo usuário	Identificador, Nome, País, Estado
Galpão	Os galpões são os locais onde as aves (Flocks) serão alocados para a criação	Identificador, Nome, Data de Construção, Tamanho, Capacidade e Cidade mais Próxima
Flock	Os flocks são um conjunto de aves que são criados pelos produtores para	Identificador, Data de Nascimento, Número de Aves Alocadas, Tipo, Raça, Sistema, Fonte de Água
Produção	A produção representa o número de ovos produzidos pelas aves	Identificador, Data de Input, Data de Início, Data Final, Ovo Jumbo, Ovo Extra Grande, Ovo Grande, Ovo Médio, Ovo Pequeno, Número Total de Ovos, Semana
Performance	Representa os dados de performance das aves.	Identificador, Data de Input, Data de Início, Data Final, Número de Aves Mortas, Quantidade de Ração Ingerida, Quantidade de Água Ingerida, Semana
Documentos	São todos os documentos relacionados aos Flocks que o produtor julga serem importantes de se armazenar	Identificador, Data de Input, Nome, Categoria, Arquivo
Dieta	Representa as fórmulas de rações que os produtores usam para alimentar as aves	Identificador, Data de Input, Nome, Empresa Provedora, Tipo de Ração, Granulometria
Inventário de Ração	Estoque de ração contida nos silos dos galpões	Identificador, Data de Início da Alimentação, Número do Silo, Quantidade, Custo por Tonelada, Custo por Transporte, Quantidade Restante

Transação	Por vezes os produtores poderiam ter um excedente de aves ou perder muitas inesperadamente, a função desta entidade é facilitar o pedido de venda e compra de aves	Identificador, Data da transação, Situação, Quantidade, Valor
-----------	--	---

4. Dicionário de Relações

Entidades Envolvidas	Restrições Estabelecidas de Cardinalidade	Restrições Estabelecidas de Participação
Produtor - Fazenda	Um produtor pode possuir N Fazendas, sendo uma Fazenda associada a 1 Produtores	Participação Total para ambos
Fazenda - Galpão	Uma Fazenda pode possuir N Galpões, sendo o Galpão associado a 1 Fazenda, necessariamente	Participação Total para ambos
Galpão - Flock	Um Galpão pode possuir 0..1 Flocks, sendo o Flock associado a 1 Galpão, necessariamente.	Participação Parcial para Galpão e Total para Flock
Performance - Inventário de Ração	Uma Performance deve possuir 1 Inventários de Ração, sendo o Inventário de Ração relacionado a N Performance, necessariamente	Participação Total para ambos
Flock - Produção	Um Flock pode possuir N Produção, sendo a Produção relacionada a 1 Flock, necessariamente	Participação Parcial para Flock e Total para Produção
Flock - Performance	Um Flock pode possuir N Performance, sendo a Performance relacionada a 1 Flock, necessariamente	Participação Parcial para Flock e Total para Performance
Flock - Documentos	Um Flock pode possuir N Documentos, sendo a Documentos relacionada a 1 Flock, necessariamente	Participação Parcial para Flock e Total para

		Documentos
Inventário de Ração - Dieta	Um Inventário pode possuir 1 Dieta, Dieta pode estar em 0...N Inventários	Participação Total para Inventário de Ração e Parcial para Dieta
Flock - Transação	Um Flock pode possuir N Transações, sendo uma Transação relacionada a 1 Flock	Participação parcial para Flock e participação total para Transação

5. Dicionário de Atributos

Produtor				
Nome do Campo	Nome Interno	Tipo	Semântica	Categorias
Username	username	String	Código de identificação	<ul style="list-style-type: none"> - Chave primária - Simples - Monovalorado - Armazenado - Conjunto de Valores: 6 <= x <= 20 dígitos
Nome	name	String	Nome do Usuário, contendo Primeiro e Último nome	<ul style="list-style-type: none"> - Monovalorado - Composto - Armazenado - Conjunto de Valores: Letras, sendo x <= 50 dígitos

Email	email	String	Email do usuário que será usado para contato	<ul style="list-style-type: none"> - Chave Candidata - Simples - Monovalorada - Armazenada
Senha	password	String	Senha do usuário	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado - Conjunto de Valores: Letras e Números, 8 <= x <= 20 dígitos

Fazenda				
Nome do Campo	Nome Interno	Tipo	Semântica	Categorias
Identificador	id	Integer	Código de identificação, gerado automaticamente ao se criar uma nova fazenda	<ul style="list-style-type: none"> - Chave Primária - Simples - Monovalorado - Armazenado
Nome	name	String	Nome da Fazenda	<ul style="list-style-type: none"> - Simples - Monovalorado

				<ul style="list-style-type: none"> - Armazenado - Conjunto de Valores: Letras e Números, 5 <= x <= 50 dígitos
País	country	String	País em que a fazenda está localizada, sendo registrado na lista somente aqueles onde o software é comercializado	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado - Conjunto de Valores: Países listados
Estado	state	String	Estado em que a fazenda está localizada. O Estado deve corresponder a lista daqueles localizados no país selecionado.	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado - Conjunto de Valores: Estados listados

Galpão				
Nome do Campo	Nome Interno	Tipo	Semântica	Categorias
Identificador	id	Integer	Código de identificação, gerado automaticamente ao se criar um novo galpão	<ul style="list-style-type: none"> - Chave Primária - Simples - Monovalorado - Armazenado
Nome	name	String	Nome do Galpão	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado - Conjunto de Valores: Letras e Números, 5 <= x <= 50 dígitos
Data de Construção	dt_build	Date: mm/dd/YYYY	Data em que o galpão foi construído	<ul style="list-style-type: none"> - Simples - Monovalorado

				<ul style="list-style-type: none"> - Armazenado - Conjunto de Valores: $x \geq$ Data de <i>input</i>
Tamanho	size	Integer	Tamanho do galpão em <i>Sqr. Ft.</i>	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado - Conjunto de Valores: $0 < x \leq 5$ dígitos
Capacidade	capacity	Integer	A capacidade máxima que um galpão pode suportar de aves	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado - Conjunto de Valores: $0 < x \leq 5$ dígitos
Cidade Próxima	nearest_city	String	É comum de uma mesma fazenda possuir galpões em localidades diferentes, então deve-se sempre especificar a cidade mais próxima aos galpões. A lista de cidades selecionáveis são aquelas que estão dentro do estado em que a fazenda relacionada ao galpão está localizada.	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado - Conjunto de Valores: Cidades listadas

Flock				
Nome do Campo	Nome Interno	Tipo	Semântica	Categorias

Identificador	id	Integer	Código de identificação, gerado automaticamente ao se cadastrar novos dados de produção	<ul style="list-style-type: none"> - Chave Primária - Simples - Monovalorado - Armazenado
Data de Nascimento	dt_hatch	Date: mm/dd/YYYY	Data em que as aves nasceram	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado - Conjunto de Valores: Deve ser menor que a data de <i>input</i>
Aves Alocadas	#_birds_housed	Integer	Quantidade de aves contidas no Flock	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado - Conjunto de Valores: $0 < x \leq 5$ dígitos
Tipo	type	String	Fase da vida em que as aves se encontram	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado - Conjunto de Valores: Layer ou Pullet
Raça	strain	String	Genética da ave	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado - Conjunto de Valores: Babcock, Dekalb ou Lohmann
Sistema	system	String	Tipo/Sistema de criação que a ave terá	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado - Conjunto de Valores: Free Range, Free Run, Enriched, Conventional ou Aviary

Fonte de Água	water_source	String	Origem da água que será usada para sustentar a necessidade das aves	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado - Conjunto de Valores: Well, City, Reverse Hosmosis
---------------	--------------	--------	---	--

Produção				
Nome do Campo	Nome Interno	Tipo	Semântica	Categorias
Identificador	id	Integer	Código de identificação, gerado automaticamente ao se cadastrar novos dados de produção	<ul style="list-style-type: none"> - Chave Primária - Simples - Monovalorado - Armazenado
Data de <i>Input</i>	dt_input	Date: mm/dd/YYYY	Data em que os dados foram inseridos	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado
Data Inicial	dt_from	Date: mm/dd/YYYY	Data em que os dados de input começaram a ser coletados	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado - Conjunto de Valores: Deve ser >= que a data de nascimento do Flock
Data Final	dt_to	Date:	Data final de coleta dos dados de input	<ul style="list-style-type: none"> - Simples

		mm/dd/YYYY		<ul style="list-style-type: none"> - Monovalorado - Armazenado - Conjunto de Valores: Deve ser \geq que a data inicial da coleta
Ovo Jumbo	egg_jumbo	Integer	Quantidade de ovos produzidos do tamanho jumbo	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado - Conjunto de Valores: $0 < x \leq 5$ dígitos
Ovo Extra Grande	egg_xl	Integer	Quantidade de ovos produzidos do tamanho extra grande	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado - Conjunto de Valores: $0 < x \leq 5$ dígitos
Ovo Grande	egg_large	Integer	Quantidade de ovos produzidos do tamanho grande	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado - Conjunto de Valores: $0 < x \leq 5$ dígitos
Ovo Médio	egg_medium	Integer	Quantidade de ovos produzidos do tamanho médio	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado - Conjunto de Valores: $0 < x \leq 5$ dígitos
Ovo Pequeno	egg_small	Integer	Quantidade de ovos produzidos do tamanho pequeno	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado - Conjunto de Valores: $0 < x \leq 5$ dígitos

Ovos Totais	egg_total	Integer	Total de ovos produzidos no período referenciado	- Simples - Monovalorado - Derivado
Semana	week	Integer	Número de semanas de vida da ave no período referenciado	- Simples - Monovalorado - Derivado

Performance				
Nome do Campo	Nome Interno	Tipo	Semântica	Categorias
Identificador	id	Integer	Código de identificação, gerado automaticamente ao se cadastrar um novo input de dados de performance	- Chave Primária - Simples - Monovalorado - Armazenado
Data de <i>Input</i>	dt_input	Date: mm/dd/YYYY	Data em que os dados foram inseridos	- Simples - Monovalorado - Armazenado - Conjunto de Valores: Deve ser >= que a data de nascimento do Flock
Data Inicial	dt_from	Date: mm/dd/YYYY	Data em que os dados de input começaram a ser coletados	- Simples - Monovalorado - Armazenado - Conjunto de Valores: Deve ser >= que a data de <i>input</i>

Data Final	dt_to	Date: mm/dd/YYYY	Data final de coleta dos dados de input	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado - Conjunto de Valores: Deve ser >= que a data inicial da coleta
Número de Aves Mortas	numbers_died	Integer	Quantidade de aves mortas no período	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado - Conjunto de Valores: $0 < x \leq 5$ dígitos
Quantidade de Ração Ingerida	feed_intake	Integer	Quantidade em gramas de ração ingerida no período	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado - Conjunto de Valores: $0 < x \leq 5$ dígitos
Quantidade de Água Ingerida	water_intake	Integer	Quantidade em mililitros de água ingerida no período	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado - Conjunto de Valores: $0 < x \leq 5$ dígitos
Semana	week	Integer	Número de semanas de vida da ave no período referenciado	<ul style="list-style-type: none"> - Simples - Monovalorado - Derivado

Documentos				
Nome do	Nome Interno	Tipo	Semântica	Categorias

Campo				
Identificador	id	Integer	Código de identificação, gerado automaticamente ao se cadastrar um novo documento	<ul style="list-style-type: none"> - Chave Primária - Simples - Monovalorado - Armazenado
Data de <i>Input</i>	dt_input	Date: mm/dd/YYYY	Data em que os dados foram inseridos	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado - Conjunto de Valores: Deve ser >= que a data de nascimento do Flock
Nome	name	String	Nome do arquivo	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado - Conjunto de Valores: Deve possuir 5 <= x <= 50 dígitos
Categoria	category	String	Categoria que o usuário poderá separar os arquivos para se filtrar com mais facilidade	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado - Conjunto de Valores: Deve possuir 5 <= x <= 20 dígitos
Arquivo	file	File	Arquivos que representam um determinado flock, que o usuário achar necessário serem arquivos	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado - Conjunto de Valores: PDF de no máximo 5 mb

Dieta

Nome do Campo	Nome Interno	Tipo	Semântica	Categorias
Identificador	id	Integer	Código de identificação, gerado automaticamente ao se cadastrar uma nova dieta	<ul style="list-style-type: none"> - Chave Primária - Simples - Monovalorado - Armazenado
Data de <i>Input</i>	dt_input	Date: mm/dd/YYYY	Data em que os dados foram inseridos, gerado automaticamente ao se cadastrar um novo dado	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado
Nome	name	String	Nome da Dieta	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado - Conjunto de Valores: Deve possuir 5 <= x <= 50 dígitos
Empresa Provedora	feed_company	String	Nome da empresa que fez a formulação da ração	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado - Conjunto de Valores: Deve possuir 5 <= x <= 50 dígitos
Tipo de Ração	feed_type	String	Composição da Ração	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado - Conjunto de Valores: Complete ou Supplement
Granulometria	granulometry	String	Granulometria da ração	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado - Conjunto de Valores: Crumble, Mash ou Pellet

Inventário de Ração				
Nome do Campo	Nome Interno	Tipo	Semântica	Categorias
Identificador	id	Integer	Código de identificação, gerado automaticamente ao se cadastrar uma nova compra de ração	<ul style="list-style-type: none"> - Chave Primária - Simples - Monovalorado - Armazenado
Data de Início da Alimentação	dt_feed_start	Date: mm/dd/YYYY	Data em que os dados foram inseridos	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado - Conjunto de Valores: Deve ser maior que a data de nascimento do Flock
Quantidade	quantity	Float	Nome da empresa que fez a formulação da ração	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado - Conjunto de Valores: $0 < x \leq 5$ dígitos
Custo por	cost_per_ton	String	Composição da Ração	<ul style="list-style-type: none"> - Simples

Tonelada				<ul style="list-style-type: none"> - Monovalorado - Armazenado - Conjunto de Valores: $0 < x \leq 5$ dígitos
Custo por Transporte	freight_cost	Float	Granulometria da ração	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado - Conjunto de Valores: $0 < x \leq 5$ dígitos
Quantidade Restante	tons_left	Float	Valor calculado através da subtração da quantidade comprada pela quantidade consumida durante o período de análise	<ul style="list-style-type: none"> - Simples - Monovalorado - Derivado

Transação				
Nome do Campo	Nome Interno	Tipo	Semântica	Categorias
Identificador	id	Integer	Código de identificação, gerado automaticamente ao se cadastrar uma nova requisição	<ul style="list-style-type: none"> - Chave Primária - Simples - Monovalorado - Armazenado
Data da Transação	dt_transaction	Date: mm/dd/YYYY	Data da transação	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado - Conjunto de Valores: Deve ser

				maior que a data de criação do Flock
Situação	situation	string	Situação do pedido realizado, podendo ser de compra ou venda	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado
Quantidade	quantity	Integer	Quantidade de galinhas vendidas/compradas	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado - Conjunto de Valores: $0 < x \leq 5$ dígitos
Custo	cost	Float	Custo total da transação	<ul style="list-style-type: none"> - Simples - Monovalorado - Armazenado - Conjunto de Valores: $0 < x \leq 5$ dígitos

8,9 / 9,0

6. Modelo Relacional

Producer (username, first_name, last_name, email, password)

Farm (id, name, country, state, nameProducer)

Farm[nameProducer] → {b} Producer[username]

Barn (id, name, dt_build, size, capacity, nearest_city, idFarm)

Barn[idFarm] → {b} Farm[id]

Flock (id, dt_hatch, birds_housed, type, strain, system, water_source, idBarn)

Flock[idBarn] → {b} Barn[id]

Document (id, name, category, file, numberFlock)

Document[numberFlock] → {b} Flock[id]

Transaction (id, dt_transaction, situation, quantity, cost, numberFlock)

Transaction[numberFlock] → {b} Flock[id]

Diet (id, name, feed_company, feed_type, granulometry, crude_protein, feed_allowance, idInventory)

Diet[idInventory] → {b} Feed Inventory[id]

bin number?

Feed Inventory (id, dt_feed_start, quantity, cost_per_ton, freight_cost, idPerformance)

Feed Inventory[idPerformance] → {b} Performance[id]

Production (id, dt_input, dt_from, dt_to, egg_jumbo, egg_large, egg_medium, egg_small, numberFlock)

Production[numberFlock] → {b} Flock[id]

Performance (id, dt_input, dt_from, dt_to, numbers_died, feed_intake, water_intake, feed_used, numberFlock)
Performance[numberFlock] → {b} Flock[id]

7. Modelo Físico

4,8 / 6,0

```
CREATE DATABASE farm_management;
```

```
CREATE TABLE Producer (  
    username    VARCHAR(20)    NOT NULL,  
    first_name  VARCHAR(30)    NOT NULL,  
    last_name   VARCHAR(30)    NOT NULL,  
    email       VARCHAR(40)    NOT NULL,  
    password    VARCHAR(40)    NOT NULL,  
    CONSTRAINT Pk_Producer PRIMARY KEY (username),  
    CONSTRAINT Min_username_size CHECK (LENGTH(username) > 6),  
    CONSTRAINT Min_email_size CHECK (LENGTH(email) > 6),  
    CONSTRAINT Min_password_size CHECK (LENGTH(password) > 8)  
);
```

```
CREATE TABLE Farm (  
    id          NUMERIC(4)      NOT NULL,  
    producer    VARCHAR(20)     NOT NULL,  
    name        VARCHAR(50)     NOT NULL,  
    country     VARCHAR(40)     NOT NULL,  
    state       VARCHAR(40)     NOT NULL,  
    CONSTRAINT Pk_Farm PRIMARY KEY (id),
```

```
CONSTRAINT Ck_country CHECK (country in ('Brasil', 'Canada')),  
CONSTRAINT Ck_state CHECK (state in ('Minas Gerais', 'Alberta')),  
CONSTRAINT Min_name_size CHECK (LENGTH(name) > 5)  
);
```

```
CREATE TABLE Barn (  
    id            NUMERIC(4)      NOT NULL,  
    farm          NUMERIC(4)      NOT NULL,  
    name          VARCHAR(50)     NOT NULL,  
    dt_build      DATE            NOT NULL,  
    size          NUMERIC(5,2)    NOT NULL,  
    capacity      NUMERIC(6)      NOT NULL,  
    nearest_city  VARCHAR(40)     NOT NULL,  
    CONSTRAINT Pk_Barn PRIMARY KEY (id),  
    CONSTRAINT Ck_nearest_city CHECK (nearest_city in ('Ouro Preto', 'Edmonton')),  
    CONSTRAINT Min_name_size CHECK (LENGTH(name) > 8)  
);
```

```
CREATE TABLE Flock (  
    id            NUMERIC(4)      NOT NULL,  
    barn          NUMERIC(4)      NOT NULL,  
    dt_hatch      DATE            NOT NULL,  
    birds_housed  NUMERIC         NOT NULL,  
    type          VARCHAR(40)     NOT NULL,  
    strain        VARCHAR(40)     NOT NULL,  
    system        VARCHAR(40)     NOT NULL,  
    water_source  VARCHAR(40)     NOT NULL,  
    CONSTRAINT Pk_Flock PRIMARY KEY (id),
```

```
CONSTRAINT Ck_type CHECK (type in ('layer','pullet')),  
CONSTRAINT Ck_strain CHECK (strain in ('babcock','dekalb', 'lohmann')),  
CONSTRAINT Ck_system CHECK (system in ('free range','free run', 'enriched', 'conventional', 'aviary'))  
);
```

```
CREATE TABLE Production (  
    id                NUMERIC(4)    NOT NULL,  
    flock             NUMERIC(4)    NOT NULL,  
    dt_input          DATE          NOT NULL,  
    dt_from           DATE          NOT NULL,  
    dt_to            DATE          NOT NULL,  
    egg_jumbo         NUMERIC(5)    NULL,  
    egg_large         NUMERIC(5)    NULL,  
    egg_medium        NUMERIC(5)    NULL,  
    egg_small         NUMERIC(5)    NULL,  
    egg_totals        NUMERIC(5)    NULL,  
    CONSTRAINT Pk_Production PRIMARY KEY (id)  
);
```

Derivado, até então.

```
CREATE TABLE Performance (  
    id                NUMERIC(4)    NOT NULL,  
    flock             NUMERIC(4)    NOT NULL,  
    feed_inventory    NUMERIC(4)    NOT NULL,  
    dt_input          DATE          NOT NULL,  
    dt_from           DATE          NOT NULL,  
    dt_to            DATE          NOT NULL,  
    number_died       NUMERIC(5)    NULL,  
    feed_intake       NUMERIC(5)    NULL,
```

Não há no relacional
proposto desta forma

```
water_intake      NUMERIC(5)      NULL,  
CONSTRAINT Pk_Performance PRIMARY KEY (id)  
);
```

```
CREATE TABLE Document (  
    id              NUMERIC(2)      NOT NULL,  
    flock           NUMERIC(4)      NOT NULL,  
    dt_input        DATE            NOT NULL,  
    name            VARCHAR(40)     NOT NULL,  
    category        VARCHAR(40)     NOT NULL,  
    arquivo         VARCHAR(500)    NOT NULL,  
    CONSTRAINT Pk_Document PRIMARY KEY (id)  
);
```

Não há no relacional

```
CREATE TABLE Diet (  
    id              NUMERIC(4)      NOT NULL,  
    dt_input        DATE            NOT NULL,  
    name            VARCHAR(40)     NOT NULL,  
    feed_company    VARCHAR(40)     NOT NULL,  
    feed_type       VARCHAR(40)     NOT NULL,  
    granulometry    VARCHAR(40)     NOT NULL,  
    CONSTRAINT Pk_Diet PRIMARY KEY (id),  
    CONSTRAINT Ck_granulometry CHECK (granulometry in ('crumble','mash', 'pellet')),  
    CONSTRAINT Ck_feed_type CHECK (feed_type in ('complete','supplement'))  
);
```

Não "casa" com
o relacional
proposto

```
CREATE TABLE FeedInventory (  
    id              NUMERIC(4)      NOT NULL,  
    flock           NUMERIC(4)      NOT NULL,  
    dt_input        DATE            NOT NULL,  
    name            VARCHAR(40)     NOT NULL,  
    category        VARCHAR(40)     NOT NULL,  
    arquivo         VARCHAR(500)    NOT NULL,  
    CONSTRAINT Pk_FeedInventory PRIMARY KEY (id)  
);
```


id	NUMERIC(4)	NOT NULL,	
diet	NUMERIC(4)	NOT NULL,	Performance?
dt_feed_start	DATE	NOT NULL,	
quantity	NUMERIC(2)	NOT NULL,	
cost_per_ton	NUMERIC(4,2)	NOT NULL,	
freight_cost	NUMERIC(4,2)	NOT NULL,	
CONSTRAINT Pk_Feed_Inventory PRIMARY KEY (id)			

);

```
CREATE TABLE Transaction (
    id                NUMERIC(2)    NOT NULL,
    flock             NUMERIC(4)    NOT NULL,
    dt_trasaction     DATE          NOT NULL,
    situation         VARCHAR(40)   NOT NULL,
    quantity          NUMERIC(4,2)  NOT NULL,
    cost              NUMERIC(4,2)  NOT NULL,
    CONSTRAINT Pk_Transaction PRIMARY KEY (id),
    CONSTRAINT Ck_situation CHECK (situation in ('Compra', 'Venda'))
);
```

```
ALTER TABLE Farm
ADD CONSTRAINT Fk_Producer FOREIGN KEY (producer)
REFERENCES Producer(username) ON UPDATE CASCADE;
```

```
ALTER TABLE Barn
ADD CONSTRAINT Fk_Farm FOREIGN KEY (farm) REFERENCES Farm(id) ON UPDATE CASCADE;
```

```
ALTER TABLE Flock
  ADD CONSTRAINT Fk_Barn FOREIGN KEY (barn) REFERENCES Barn(id) ON UPDATE CASCADE;
```

```
ALTER TABLE Production
  ADD CONSTRAINT Fk_Flock FOREIGN KEY (flock) REFERENCES Flock(id) ON UPDATE CASCADE;
```

```
ALTER TABLE Performance
  ADD CONSTRAINT Fk_Flock FOREIGN KEY (flock) REFERENCES Flock(id) ON UPDATE CASCADE;
```

```
ALTER TABLE Performance
  ADD CONSTRAINT Fk_Feed_Inventory FOREIGN KEY (feed_inventory) REFERENCES FeedInventory(id) ON UPDATE CASCADE;
```

```
ALTER TABLE Document
  ADD CONSTRAINT Fk_Flock FOREIGN KEY (flock) REFERENCES Flock(id) ON UPDATE CASCADE;
```

```
ALTER TABLE Transaction
  ADD CONSTRAINT Fk_Flock FOREIGN KEY (flock) REFERENCES Flock(id) ON UPDATE CASCADE;
```

```
ALTER TABLE FeedInventory
  ADD CONSTRAINT Fk_Diet FOREIGN KEY (diet) REFERENCES Diet(id) ON UPDATE CASCADE;
```

```
INSERT INTO Producer
  VALUES ('Niquini', 'Gabriel', 'Niquini', 'Gabriel.Niquini@aluno.ufop.edu.br', '0123456789');
```

```
INSERT INTO Farm
  VALUES(1, 'Niquini', 'Fazendinha do barulho', 'Brasil', 'Minas Gerais');
```

Não há no relacional proposto

No relacional proposto, era com Performance e não Diet

No relacional, há tal restrição que não apareceu no script:
Diet[idInventory] → {b} Feed Inventory[id]

```
INSERT INTO Barn  
VALUES(1,1,'Silo da doidera', '12/06/01', 450, 900, 'Ouro Preto');
```

```
INSERT INTO Flock  
VALUES(1,1, '07/14/01', 300, 'layer', 'babcock', 'free run', 'itacity');
```

```
INSERT INTO Production  
VALUES(1, 1, '07/14/01', '07/15/01', '08/18/01');
```

```
INSERT INTO Document  
VALUES (1, 1, '07/14/01', 'Documento serelepe', 'pdf', 'documentoSerelepe.pdf');
```

```
INSERT INTO Diet  
VALUES (1,'08/14/01', 'Dietinha boa', 'RomuloLTDA', 'complete', 'mash');
```

```
INSERT INTO FeedInventory  
VALUES(1,1, '08/14/01', 50, 25.5, 31.2);
```

```
INSERT INTO Performance  
VALUES(1,1, 1, '07/14/01', '07/15/01', '08/18/01');
```

```
INSERT INTO Transaction  
VALUES (1, 1, '05/15/09', 'Compra', 10.00, 35.00);
```

Item 4 do trabalho: 5,5 / 6,0

**=> Faltou mais "entrada de dados" nas
consultas solicitadas**