



**Departamento de Ciência da Computação
UFLA - Universidade Federal de Lavras
GCC263 - Introdução a Sistemas de Banco de Dados
Prof. Denilson Alves Pereira**

Trabalho Prático – Etapa Final

GRUPO 7

André Araújo Mendonça - 202310982
Gustavo Henrique Moraes Filho - 202310429
Lilian Carla De Freitas - 202310885
Samuel Moreira Abreu - 202220673
Tiago De Paula Martins - 202311001

1. INTRODUÇÃO

O objetivo deste trabalho é fazer o levantamento de requisitos de dados para um jogo envolvendo o universo Pokémons. Neste projeto, aplicamos os princípios fundamentais de gerenciamento de banco de dados na prática. O desafio consiste em desenvolver um sistema de banco de dados que representa um sistema de gerenciamento de Pokémons em um jogo.

No módulo 1 é apresentado um levantamento de requisitos de dados de acordo com as regras do Modelo de Entidades e Relacionamentos (ER), seguindo a notação do livro texto utilizado na disciplina e utilizando a ferramenta TerraER.

Já no módulo 2, é apresentado o Diagrama relacional, incluindo as restrições de integridade referencial com opção de exclusão. Utilizamos a ferramenta MySQL Workbench para fazer o diagrama relacional. Também adicionamos o Dicionário de dados. Semelhante ao dicionário de dados entregue na Etapa 1, porém organizado por tabela ao invés de tipo entidade/relacionamento. Além de realizarmos a Normalização de dados do nosso projeto.

Por fim, no módulo 3 o objetivo foi implementar o banco de dados definido nas Etapas 1 e 2 em um SGBD relacional e criar um exemplo de interface Web com o usuário em PHP. Criamos as tabelas, restrições de integridade, procedimentos, triggers e realizamos as consultas, inserções, modificações e exclusões de dados no sistema.

2. DESCRIÇÃO DO PROBLEMA

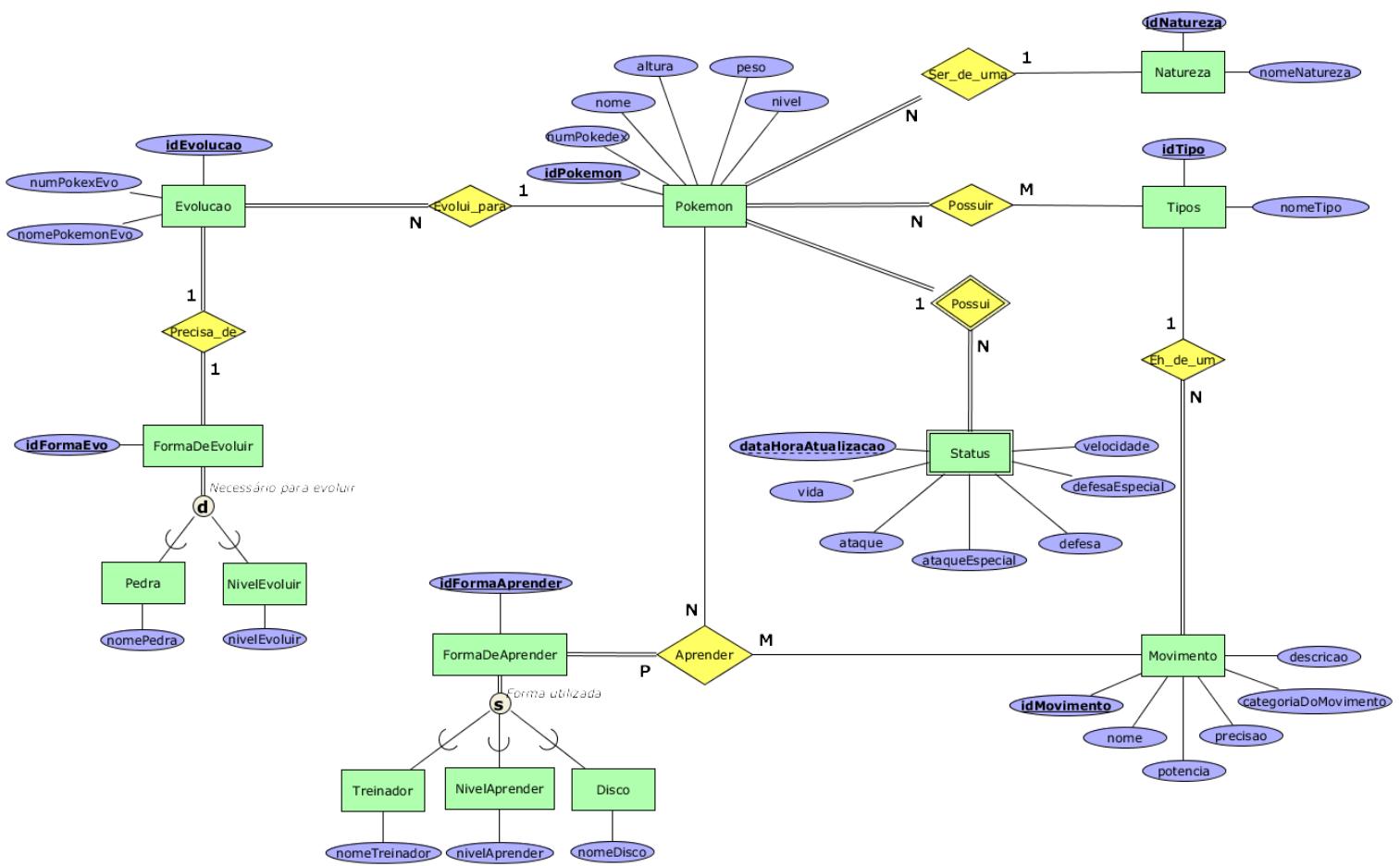
O problema consiste em criar um sistema de gerenciamento de Pokémons. No universo do anime Pokémon, essas criaturas se assemelham a animais que lutam entre si, possuindo características próprias focadas em combate e crescimento (evoluções). Cada pokémon é único em ID e existe um número definido na pokédex para cada pokémon, mas podem existir vários pokémons que se chamam caterpie e possuem o número de pokédex 10, mas seus ID são distintos.

Os Pokémons podem evoluir para outros Pokémons das seguintes maneiras: por meio de aumento de nível ou por meio de uso de pedras evolutivas. Um único pokémon pode evoluir para zero ou várias evoluções, cada evolução variando na forma de evoluir e possuindo um nome novo e um número na pokédex novo. Eles também pertencem a um ou mais tipos, como normal, água, fogo, inseto, dragão, pedra, entre outros. Esses tipos também são atribuídos aos seus movimentos (similares a poderes ou magias). Cada movimento pertence a apenas um tipo, mas existem diversos movimentos de um mesmo tipo. Esses movimentos podem ser aprendidos de diferentes formas(únicas ou em conjunto): ao subir de nível, através de discos de movimentos TMs/HMs(absorvendo o poder desse disco) e por meio de treinamento realizado por treinadores específicos, como se fosse um tutor.

Como os Pokémons estão envolvidos em batalhas, é necessário definir parâmetros para medir sua força, vida e resistência. Esses parâmetros são chamados de status e podem mudar constantemente quando um Pokémon evolui ou sobe de nível. A natureza do Pokémon, que é determinada ao nascer e não pode ser alterada, define a sua personalidade e forma de se relacionar com outros Pokémons, por exemplo, se ele for calm, relaxed, seriuos, lonely, entre outros!

Este trabalho de banco de dados sobre Pokémons visa criar um sistema robusto que permita o gerenciamento dessas diversas características, facilitando a organização e o acesso a informações detalhadas sobre cada Pokémon, suas evoluções, tipos, movimentos e status.

3. DIAGRAMA ER



4. DICIONÁRIO DE DADOS

ENTIDADES:

Entidade: Pokemon

Descrição: Representa uma instância de um Pokémon do sistema.

Nome	Descrição	Dominio	Permite nulo? (S/N)
idPokemon	Identificação única de um Pokémon	Números inteiros positivos de três dígitos (de 001 a 999)	N
nome	Nome do Pokémon	Texto (40)	N
altura	Altura de um Pokémon	Números racionais positivos de até 2 dígitos com 2 casas decimais no formato XX.XX	N
peso	Peso de um Pokémon	Números racionais positivos de até 4 dígitos com 2 casas decimais no formato XX.XX	N
nivel	Nível do Pokémon	Números inteiros positivos de três dígitos (de 001 a 100)	N
numPokedex	Número da pokédex de um Pokémon	Números inteiros positivos de três dígitos (de 001 a 999)	N

Entidade: Status**Descrição:** Status de um pokémon

Nome	Descrição	Domínio	Permite nulo? (S/N)
dataHoraAtualizacao	Data e hora da última atualização dos status de um pokémon.	Data no formato AAAA/MM/DD e HH/MM	N
vida	Quantidade de vida de um pokémon	Números inteiros positivos de três dígitos(de 000 a 999)	N
ataque	Dano físico de um pokémon	Números inteiros positivos de três dígitos (de 000 a 999)	N
defesa	Resistência a danos físicos de um pokémon	Números inteiros positivos de três dígitos (de 000 a 999)	N
defesaEspecial	Resistência a danos especiais de um pokémon	Números inteiros positivos de três dígitos (de 000 a 999)	N
ataqueEspecial	Dano especiais de um pokémon	Números inteiros positivos de três dígitos (de 000 a 999)	N
velocidade	Velocidade de um pokémon	Números inteiros positivos de três dígitos (de 000 a 999)	N

Entidade:Tipos**Descrição:** Define os tipos que um pokémon pertence.

Nome	Descrição	Domínio	Permite nulo? (S/N)
idTipo	Identificação do tipo	Números inteiros positivos de dois dígitos (de 01 a 99)	N
nomeTipo	Nome dos tipos (elétrico, terra, etc...)	Texto (40)	N

Entidade:FormaDeEvoluir**Descrição:** Forma que um pokémon possui para se transformar (evoluir) em outro, pode se evoluir de duas formas.

Nome	Descrição	Domínio	Permite nulo? (S/N)
idFormaEvo	Identificação da forma de evolução	Números inteiros positivos de três dígitos (001 a 999)	N

Entidade:Pedra**Descrição:** Uma das formas de evolução

Nome	Descrição	Domínio	Permite nulo? (S/N)
nomePedra	Nome da pedra	Texto (40)	S

Entidade:NivelEvoluir**Descrição:** Uma das formas de evolução

Nome	Descrição	Domínio	Permite nulo? (S/N)
nívelEvoluir	nível necessário para a evolução	Números inteiros positivos de três dígitos (de 001 a 100)	S

Entidade:Movimento**Descrição:** Habilidades, magias e poderes que um pokémon pode ter.

Nome	Descrição	Domínio	Permite nulo? (S/N)
idMovimento	Identificação do movimento	Números inteiros positivos de três dígitos (de 001 a 999)	N
nome	Nome do movimento (habilidade)	Texto (40)	N
potencia	Dano do movimento	Números inteiros positivos de três dígitos (de 000 a 999)	N
precisao	Chance de acerto de um movimento	Números inteiros positivos de três dígitos (de 000 a 100)	N
categoriaDoMovimento	A categoria do movimento(especial, físico ou de status)	Caractere ('E' – especial, 'F' – físico e 'S' – status)	N
descricao	Descrição do movimento	Texto (200)	N

Entidade:Natureza**Descrição:** Personalidade e como o pokémon se relaciona com outros.

Nome	Descrição	Domínio	Permite nulo? (S/N)
idNatureza	Identificação da natureza	Números inteiros positivos de dois dígitos (de 01 a 99)	N
nomeNatureza	Nome das naturezas (timid, calm, relaxed...)	Texto (40)	N

Entidade: Evolução**Descrição:** Como um pokémon se transforma em outro.

Nome	Descrição	Domínio	Permite nulo? (S/N)
idEvolucao	identificação da evolução de um pokémon	Números inteiros positivos de três dígitos (de 001 a 999)	
numPokedex Evo	Número da pokédex que o pokémon irá evoluir	Números inteiros positivos de três dígitos (de 001 a 999)	N
nomePokemo nEvo	Nome do pokémon que ele se transformará	Texto (80)	N

Entidade: FormaDeAprender**Descrição:** Como os pokémons aprendem os movimentos.

Nome	Descrição	Domínio	Permite nulo? (S/N)
idFormaAprender	identificação da forma de aprendizado	Números inteiros positivos de três dígitos (de 001 a 999)	N

Entidade: Treinador**Descrição:** Um treinador/tutor de pokémons ensina um movimento ao pokémon.

Nome	Descrição	Domínio	Permite nulo? (S/N)
nomeTreinad or	Nome do Treinador que irá ensinar o movimento ao pokémon	Texto (40)	S

Entidade: NivelAprender

Descrição: Nível que o pokémon precisa estar para aprender esse ataque de forma natural.

Nome	Descrição	Domínio	Permite nulo? (S/N)
nivelAprender	Nível específico que o pokémon precisa estar para aprender	Números inteiros positivos de três dígitos (de 001 a 100)	S

Entidade: Disco

Descrição: Nome do disco necessário para aprender um movimento.

Nome	Descrição	Domínio	Permite nulo? (S/N)
nomeDisco	Nome do disco que o pokémon irá absorver	Texto (40)	S

RELACIONAMENTOS:

Tipo do Relacionamento :Ser_de_uma

Descrição:Um pokémon deve ser de uma e apenas uma natureza.

Nome	Descrição	Domínio	Permite nulo? (S/N)
N/A	N/A	N/A	N/A

Tipo do Relacionamento :Evolui_para

Descrição:Um pokémon pode evoluir para várias evoluções, mas existem pokémons que não possuem evoluções.

Nome	Descrição	Domínio	Permite nulo? (S/N)
N/A	N/A	N/A	N/A

Tipo do Relacionamento :Aprender

Descrição: Um Pokémon pode aprender zero ou vários movimentos, através das formas de aprendizado.

Nome	Descrição	Domínio	Permite nulo? (S/N)
N/A	N/A	N/A	N/A

Tipo do Relacionamento :Possui

Descrição:Um pokémon possui status que podem variar com o nível e evoluções.

Nome	Descrição	Domínio	Permite nulo? (S/N)
N/A	N/A	N/A	N/A

Tipo do Relacionamento :Possuir

Descrição:Um pokémon deve possuir um ou mais tipos.

Nome	Descrição	Domínio	Permite nulo? (S/N)
N/A	N/A	N/A	N/A

Tipo do Relacionamento :Eh_de_um

Descrição: Um movimento deve ser de um e apenas um tipo.

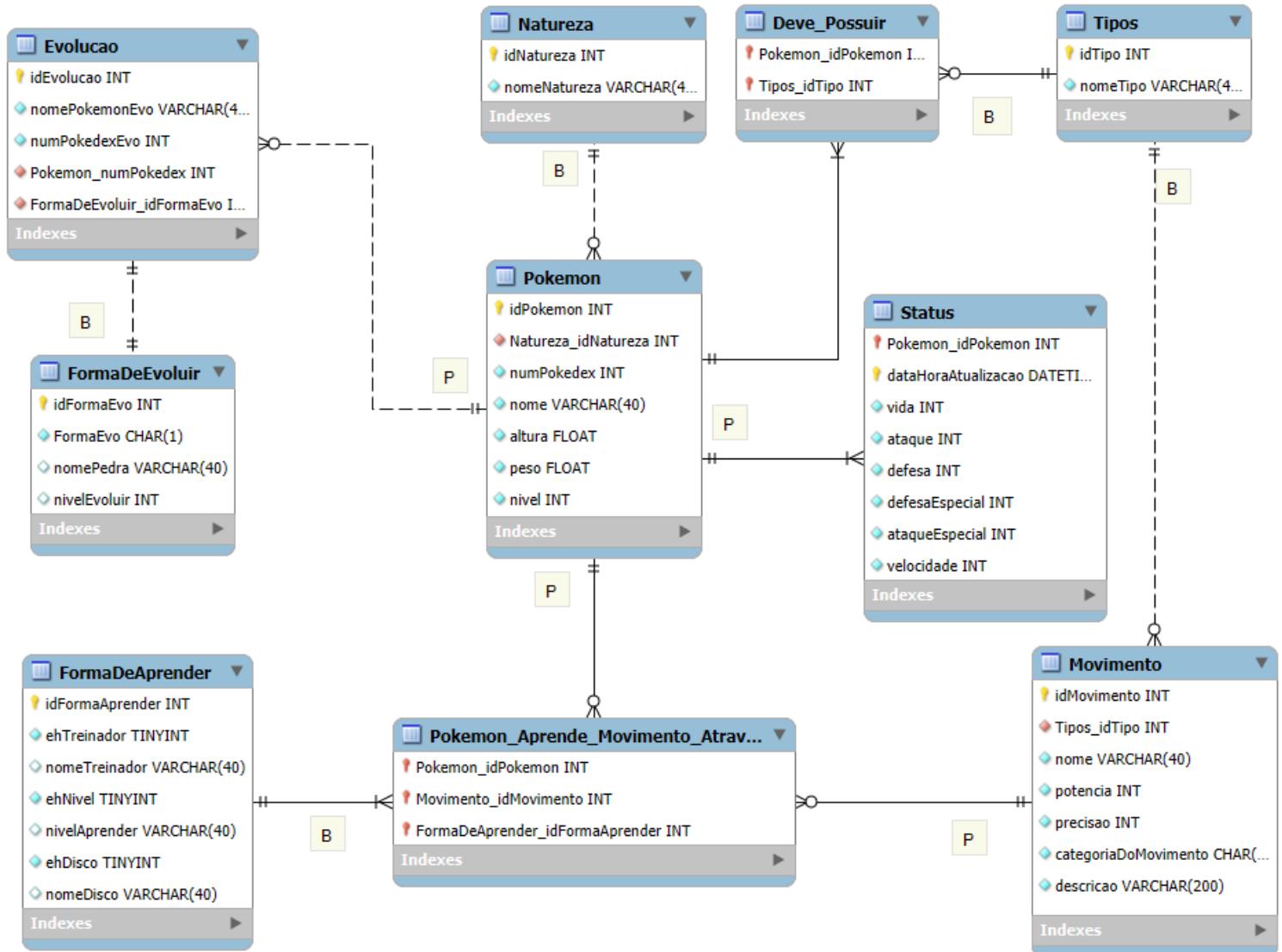
Nome	Descrição	Domínio	Permite nulo? (S/N)
N/A	N/A	N/A	N/A

Tipo do Relacionamento :Precisa_de

Descrição: Um pokémon precisa de uma condição para evoluir para outro pokémon.

Nome	Descrição	Domínio	Permite nulo? (S/N)
N/A	N/A	N/A	N/A

5. Diagrama Relacional



6. Dicionário de dados - Relacional

Tabela	Pokemon
Descrição	Pokémons do sistema.
Atributos	
Nome	Descrição
idPokemon	Número de identificação do pokémon. Autoincrementável
nome	Nome do pokémon
altura	Altura do pokémon
peso	Peso de um Pokémon
nível	Nível do Pokémon
numPokedex	Número da pokédex de um Pokémon
Natureza_idNatureza	Natureza de um Pokémon

Tabela	Status
Descrição	Status dos pokémons do sistema
Atributos	
Nome	Descrição
Pokemon_idPokemon	Chave estrangeira que referencia um pokémon.
dataHoraAtualizacao	Data e horário da última atualização dos status. Chave parcial.
ataque	Dano físico de um pokémon
defesa	Resistência a danos físicos de um pokémon
defesaEspecial	Resistência a danos especiais de um pokémon
ataqueEspecial	Dano especiais de um pokémon
velocidade	Velocidade de um pokémon
vida	Quantidade de vida de um pokémon

Tabela	Tipos
Descrição	Tipos possíveis e suas características
Atributos	
Nome	Descrição
idTipo	Identificação do tipo. Autoincrementável
nomeTipo	Nome do tipo

Tabela	Deve_Possuir
Descrição	Pokémons e seu respectivo tipo
Atributos	
Nome	Descrição
Pokemon_idPokemon	Chave estrangeira de Pokemon.
Tipos_idTipo	Chave estrangeira de Tipos.

Tabela	Evolucao
Descrição	Evoluçãoes dos pokémons e suas características
Atributos	
Nome	Descrição
idEvolucao	Identificação da evolução. Autoincrementável
nomePokemonEvo	Nome do pokémon que será obtido após evoluir.
numPokedexEvo	Número da pokédex que o pokémon base irá evoluir.
Pokemon_numPokedex	Número da pokédex do pokémon pré-evoluir.
FormaDeEvoluir_idFormaEvo	Referência de atributo de FormaDeEvoluir. Informa o método utilizado para evoluir.

Tabela	FormaDeEvoluir
Descrição	Formas de um pokémon evoluir
Atributos	
Nome	Descrição
idFormEvo	Identificação da forma de evolução. Autoincrementável
formaEvo	Informa se será por pedra(P) ou nível(N).
nomePedra	Nome da pedra utilizada para evoluir.
nivelEvoluir	Nível necessário para evoluir

Tabela	Natureza
Descrição	Todas naturezas possíveis de um pokémon ter
Atributos	
Nome	Descrição
idNatureza	Identificação única de uma natureza. Autoincrementável
nomeNatureza	Nome da natureza

Tabela	FormaDeAprender
Descrição	A forma em que um pokémon aprende determinado movimento
Atributos	
Nome	Descrição
idFormAprender	Identificação da forma de aprender. Autoincrementável
ehTreinador	Booleano que indica se será ensinado o movimento por um treinador
nomeTreinador	Nome do treinador que irá ensinar ao pokémon o movimento
ehNivel	Booleano que indica se será aprendido um movimento ao chegar em certo nível
nivelAprender	Nível necessário para o pokémon aprender determinado movimento
ehDisco	Booleano que indica se será aprendido o movimento ao absorver um disco
nomeDisco	Nome do disco que o pokémon irá absorver para aprender o movimento

Tabela	Pokemon_Aprende_Movimento_Atravez
Descrição	Relaciona os pokemons com os movimentos e forma de aprender.
Atributos	
Nome	Descrição
Pokemon_idPokemon	Chave estrangeira de Pokémon.
Movimento_idMovimento	Chave estrangeira de Movimento.
FormaDeAprender_idFormaAprender	Chave estrangeira de FormaDeAprender.

Tabela	Movimento
Descrição	Característica de um movimento
Atributos	
Nome	Descrição
idMovimento	Identificação de um movimento. Autoincrementável
Tipos_idTipo	Chave estrangeira de Tipos.
nome	Nome de um movimento
potencia	Potência de um movimento
precisao	Precisão de um movimento
categoriaDoMovimento	Categoria de um movimento
descricao	Descrição de um movimento

7. Descrição dos comandos dos Scripts SQL

Abaixo são apresentados os scripts do SQL de acordo com o que foi solicitado para a 3^a etapa do trabalho dos exercícios da letra “A” até a letra “J”.

a) Criação das tabelas

Cada bloco cria uma tabela com suas respectivas colunas e define chaves primárias e estrangeiras:

```
1 -- Desativa as verificações de unicidade de chave e integridade referencial temporariamente
2 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
3 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
4
5 -- Define o modo SQL, incluindo restrições de agrupamento e validações de dados
6 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
7
8 --
9 -- Cria o esquema 'Pokedex', se não existir
10 --
11 CREATE SCHEMA IF NOT EXISTS `Pokedex` DEFAULT CHARACTER SET utf8 ;
12 USE `Pokedex` ; -- Define o esquema Pokedex como o esquema ativo
13
```



```
1 -- -----
2 -- Tabela 'Natureza': Armazena as naturezas dos Pokémon
3 --
4 CREATE TABLE IF NOT EXISTS `Pokedex`.`Natureza` (
5   `idNatureza` INT NOT NULL AUTO_INCREMENT, -- Identificador único para a natureza, gerado automaticamente
6   `nomeNatureza` VARCHAR(40) NOT NULL, -- Nome da natureza, limitado a 40 caracteres
7   PRIMARY KEY (`idNatureza`), -- Define idNatureza como chave primária
8   ENGINE = InnoDB; -- Define o mecanismo de armazenamento como InnoDB, que suporta transações e integridade referencial
9
```



```
1 -- -----
2 -- Tabela 'Pokemon': Armazena os Pokémon e suas características básicas
3 --
4 CREATE TABLE IF NOT EXISTS `Pokedex`.`Pokemon` (
5   `idPokemon` INT NOT NULL AUTO_INCREMENT, -- Identificador único do Pokémon, gerado automaticamente
6   `Natureza_idNatureza` INT NOT NULL, -- Referência à tabela Natureza, para associar uma natureza ao Pokémon
7   `numPokedex` INT NOT NULL, -- Número do Pokémon na Pokédex
8   `nome` VARCHAR(40) NOT NULL, -- Nome do Pokémon, limitado a 40 caracteres
9   `altura` FLOAT NOT NULL, -- Altura do Pokémon
10  `peso` FLOAT NOT NULL, -- Peso do Pokémon
11  `nivel` INT NOT NULL, -- Nível do Pokémon
12  PRIMARY KEY (`idPokemon`), -- Define idPokemon como chave primária
13  INDEX `fk_Pokemon_Natureza1_idx` (`Natureza_idNatureza` ASC) VISIBLE, -- Cria um índice para a coluna Natureza_idNatureza, melhorando a performance de consultas
14  CONSTRAINT `fk_Pokemon_Natureza1` -- Define uma chave estrangeira que liga Natureza_idNatureza à coluna idNatureza na tabela Natureza
15  FOREIGN KEY (`Natureza_idNatureza`)
16  REFERENCES `Pokedex`.`Natureza`(`idNatureza`)
17  ON DELETE RESTRICT -- Restringe a exclusão de uma natureza se houver um Pokémon relacionado
18  ON UPDATE CASCADE -- Atualiza automaticamente Natureza_idNatureza se idNatureza for alterado
19  ENGINE = InnoDB;
```

```
1 -- -----
2 -- Tabela 'Status': Armazena os status dos Pokémons em um determinado momento
3 --
4 CREATE TABLE IF NOT EXISTS `Pokedex`.`Status` (
5   `Pokemon_idPokemon` INT NOT NULL,          -- Referência ao Pokémon associado a este status
6   `dataHoraAtualizacao` DATETIME NOT NULL,    -- Data e hora da atualização do status
7   `vida` INT NOT NULL,                        -- Vida do Pokémon
8   `ataque` INT NOT NULL,                      -- Ataque do Pokémon
9   `defesa` INT NOT NULL,                      -- Defesa do Pokémon
10  `defesaEspecial` INT NOT NULL,             -- Defesa Especial do Pokémon
11  `ataqueEspecial` INT NOT NULL,             -- Ataque Especial do Pokémon
12  `velocidade` INT NOT NULL,                 -- Velocidade do Pokémon
13  INDEX `fk_Status_Pokemon1_idx` (`Pokemon_idPokemon` ASC) VISIBLE, -- Cria um índice para a coluna Pokemon_idPokemon
14  PRIMARY KEY (`Pokemon_idPokemon`, `dataHoraAtualizacao`), -- Define uma chave primária composta por Pokemon_idPokemon e dataHoraAtualizacao
15  CONSTRAINT `fk_Status_Pokemon1`                -- Define uma chave estrangeira que liga Pokemon_idPokemon à coluna idPokemon na tabela Pokemon
16    FOREIGN KEY (`Pokemon_idPokemon`)
17    REFERENCES `Pokedex`.`Pokemon` (`idPokemon`)
18  ON DELETE CASCADE                           -- Exclui automaticamente os status se o Pokémon associado for deletado
19  ON UPDATE CASCADE                          -- Atualiza automaticamente Pokemon_idPokemon se idPokemon for alterado
20 ENGINE = InnoDB;
```

```
1 -- -----
2 -- Tabela 'Tipos': Armazena os diferentes tipos de Pokémons
3 --
4 CREATE TABLE IF NOT EXISTS `Pokedex`.`Tipos` (
5   `idTipo` INT NOT NULL AUTO_INCREMENT,        -- Identificador único para o tipo, gerado automaticamente
6   `nomeTipo` VARCHAR(40) NOT NULL,              -- Nome do tipo, limitado a 40 caracteres
7   PRIMARY KEY (`idTipo`),                      -- Define idTipo como chave primária
8   ENGINE = InnoDB;
9 --
10 --
11 -- Tabela 'FormaDeEvoluir': Armazena as diferentes formas de evolução dos Pokémons
12 --
13 CREATE TABLE IF NOT EXISTS `Pokedex`.`FormaDeEvoluir` (
14   `idFormaEvo` INT NOT NULL AUTO_INCREMENT,    -- Identificador único para a forma de evolução, gerado automaticamente
15   `FormaEvo` CHAR(1) NOT NULL,                  -- Tipo de forma de evolução (usando um único caractere, ex: 'L' para level, 'S' para stone)
16   `nomePedra` VARCHAR(40) NULL,                -- Nome da pedra usada para evoluir, se aplicável
17   `nivelEvoluir` INT NULL,                     -- Nível necessário para evolução, se aplicável
18   PRIMARY KEY (`idFormaEvo`),                  -- Define idFormaEvo como chave primária
19   ENGINE = InnoDB;
```

```
1 -- -----
2 -- Tabela 'Movimento': Armazena os movimentos que um Pokémon pode aprender
3 --
4 CREATE TABLE IF NOT EXISTS `Pokedex`.`Movimento` (
5   `idMovimento` INT NOT NULL AUTO_INCREMENT,      -- Identificador único para o movimento, gerado automaticamente
6   `Tipos_idTipo` INT NOT NULL,                    -- Referência ao tipo do movimento, associando à tabela Tipos
7   `nome` VARCHAR(40) NOT NULL,                    -- Nome do movimento, limitado a 40 caracteres
8   `potencia` INT NOT NULL,                       -- Potência do movimento
9   `precisao` INT NOT NULL,                        -- Precisão do movimento
10  `categoriaDoMovimento` CHAR(1) NOT NULL,       -- Categoria do movimento (ex: 'F' para físico, 'E' para especial, 'S' para status)
11  `descricao` VARCHAR(200) NOT NULL,              -- Descrição detalhada do movimento
12  PRIMARY KEY (`idMovimento`),                  -- Define idMovimento como chave primária
13  INDEX `fk_Movimento_Tipos1_idx` (`Tipos_idTipo` ASC) VISIBLE, -- Cria um índice para a coluna Tipos_idTipo
14  CONSTRAINT `fk_Movimento_Tipos1`                -- Define uma chave estrangeira que liga Tipos_idTipo à coluna idTipo na tabela Tipos
15    FOREIGN KEY (`Tipos_idTipo`)
16    REFERENCES `Pokedex`.`Tipos` (`idTipo`)
17  ON DELETE RESTRICT                            -- Restringe a exclusão de um tipo se houver movimentos associados
18  ON UPDATE CASCADE                          -- Atualiza automaticamente Tipos_idTipo se idTipo for alterado
19 ENGINE = InnoDB;
```

```
1 -- -----
2 -- Tabela 'FormaDeAprender': Armazena as formas pelas quais um Pokémon pode aprender um movimento
3 --
4 CREATE TABLE IF NOT EXISTS `Pokedex`.`FormaDeAprender` (
5   `idFormaAprender` INT NOT NULL AUTO_INCREMENT,          -- Identificador único para a forma de aprendizado, gerado automaticamente
6   `ehTreinador` TINYINT NOT NULL,                         -- Indica se a forma de aprendizado envolve um treinador (1 para sim, 0 para não)
7   `nomeTreinador` VARCHAR(40) NULL,                      -- Nome do treinador, se aplicável
8   `ehNivel` TINYINT NOT NULL,                            -- Indica se a forma de aprendizado envolve atingir um certo nível (1 para sim, 0 para não)
9   `nivelAprender` VARCHAR(40) NULL,                     -- Nível no qual o movimento é aprendido, se aplicável
10  `ehDisco` TINYINT NOT NULL,                           -- Indica se a forma de aprendizado envolve o uso de um disco TM/HM (1 para sim, 0 para não)
11  `nomeDisco` VARCHAR(40) NULL,                         -- Nome do disco TM/HM, se aplicável
12  PRIMARY KEY (`idFormaAprender`),                      -- Define idFormaAprender como chave primária
13  ENGINE = InnoDB;
```

```
1 -- -----
2 -- Tabela 'Deve_Possuir': Relação muitos-para-muitos entre Pokémon e Tipos
3 --
4 CREATE TABLE IF NOT EXISTS `Pokedex`.`Deve_Possuir` (
5   `Pokemon_idPokemon` INT NOT NULL,                      -- Referência ao Pokémon na relação
6   `Tipos_idTipo` INT NOT NULL,                          -- Referência ao tipo na relação
7   INDEX `fk_Pokemon_has_Tipos_Tipos1_idx`(`Tipos_idTipo` ASC) VISIBLE, -- Cria um índice para Tipos_idTipo
8   INDEX `fk_Pokemon_has_Tipos_Pokemon1_idx`(`Pokemon_idPokemon` ASC) VISIBLE, -- Cria um índice para Pokemon_idPokemon
9   PRIMARY KEY (`Pokemon_idPokemon`, `Tipos_idTipo`),    -- Define uma chave primária composta por Pokemon_idPokemon e Tipos_idTipo
10  CONSTRAINT `fk_Pokemon_has_Tipos_Pokemon1`           -- Define uma chave estrangeira que liga Pokemon_idPokemon à coluna idPokemon na tabela Pokemon
11  FOREIGN KEY (`Pokemon_idPokemon`)
12  REFERENCES `Pokedex`.`Pokemon`(`idPokemon`)
13  ON DELETE CASCADE,                                    -- Exclui automaticamente a relação se o Pokémon for deletado
14  ON UPDATE CASCADE,                                    -- Atualiza automaticamente Pokemon_idPokemon se idPokemon for alterado
15  CONSTRAINT `fk_Pokemon_has_Tipos_Tipos1`             -- Define uma chave estrangeira que liga Tipos_idTipo à coluna idTipo na tabela Tipos
16  FOREIGN KEY (`Tipos_idTipo`)
17  REFERENCES `Pokedex`.`Tipos`(`idTipo`)
18  ON DELETE CASCADE,                                    -- Exclui automaticamente a relação se o tipo for deletado
19  ON UPDATE CASCADE,                                    -- Atualiza automaticamente Tipos_idTipo se idTipo for alterado
20  ENGINE = InnoDB;
```

```
1 -- -----
2 -- Tabela 'Evolucao': Armazena os relacionamentos de evolução entre os Pokémons
3 --
4 CREATE TABLE IF NOT EXISTS `Pokedex`.`Evolucao` (
5   `numPokedexEvo` INT NOT NULL,          -- Número do Pokémon que está sendo evoluído
6   `Pokemon_numPokedex` INT NOT NULL,      -- Número do Pokémon resultante da evolução
7   `FormaDeEvoluir_idFormaEvo` INT NOT NULL, -- Referência à forma específica de evolução
8   INDEX `fk_Evolucao_Pokemon1_idx` (`Pokemon_numPokedex` ASC) VISIBLE, -- Cria um índice para Pokemon_numPokedex
9   INDEX `fk_Evolucao_FormadeEvoluir_idx` (`FormaDeEvoluir_idFormaEvo` ASC) VISIBLE, -- Cria um índice para FormaDeEvoluir_idFormaEvo
10  PRIMARY KEY (`numPokedexEvo`, `Pokemon_numPokedex`, `FormaDeEvoluir_idFormaEvo`), -- Define uma chave primária composta por numPokedexEvo, Pokemon_numPokedex, e FormaDeEvoluir_idFormaEvo
11  CONSTRAINT `fk_Evolucao_Pokemon1`          -- Define uma chave estrangeira que liga Pokemon_numPokedex à coluna idPokemon na tabela Pokemon
12    FOREIGN KEY (`Pokemon_numPokedex`)
13      REFERENCES `Pokedex`.`Pokemon` (`idPokemon`)
14  ON DELETE RESTRICT                      -- Restringe a exclusão se houver evoluções associadas
15  ON UPDATE CASCADE,                      -- Atualiza automaticamente Pokemon_numPokedex se idPokemon for alterado
16  CONSTRAINT `fk_Evolucao_FormadeEvoluir1` -- Define uma chave estrangeira que liga FormaDeEvoluir_idFormaEvo à coluna idFormaEvo na tabela FormaDeEvoluir
17    FOREIGN KEY (`FormaDeEvoluir_idFormaEvo`)
18      REFERENCES `Pokedex`.`FormadeEvoluir` (`idFormaEvo`)
19  ON DELETE RESTRICT                      -- Restringe a exclusão se houver evoluções associadas
20  ON UPDATE CASCADE,                      -- Atualiza automaticamente FormaDeEvoluir_idFormaEvo se idFormaEvo for alterado
21 ENGINE = InnoDB;
```

```
1 -- -----
2 -- Tabela 'Aprender': Relaciona Pokémons, movimentos e a forma como esses movimentos são aprendidos
3 --
4 CREATE TABLE IF NOT EXISTS `Pokedex`.`Aprender` (
5   `Pokemon_idPokemon` INT NOT NULL,        -- Referência ao Pokemon que aprende o movimento
6   `Movimento_idMovimento` INT NOT NULL,     -- Referência ao movimento que é aprendido
7   `FormadeAprender_idFormaAprender` INT NOT NULL, -- Referência à forma de aprendizado do movimento
8   INDEX `fk_Pokemon_has_Movimento_Movimento1_idx` (`Movimento_idMovimento` ASC) VISIBLE, -- Cria um índice para Movimento_idMovimento
9   INDEX `fk_Pokemon_has_Movimento_Pokemon1_idx` (`Pokemon_idPokemon` ASC) VISIBLE, -- Cria um índice para Pokemon_idPokemon
10  INDEX `fk_Pokemon_has_Movimento_FormadeAprender1_idx` (`FormadeAprender_idFormaAprender` ASC) VISIBLE, -- Cria um índice para FormadeAprender_idFormaAprender
11  PRIMARY KEY (`Pokemon_idPokemon`, `Movimento_idMovimento`, `FormadeAprender_idFormaAprender`), -- Define uma chave primária composta por Pokemon_idPokemon, Movimento_idMovimento, e FormadeAprender_idFormaAprender
12  CONSTRAINT `fk_Pokemon_has_Movimento_Pokemon1` -- Define uma chave estrangeira que liga Pokemon_idPokemon à coluna idPokemon na tabela Pokemon
13    FOREIGN KEY (`Pokemon_idPokemon`)
14      REFERENCES `Pokedex`.`Pokemon` (`idPokemon`)
15  ON DELETE RESTRICT                      -- Restringe a exclusão se houver associações a movimentos
16  ON UPDATE CASCADE,                      -- Atualiza automaticamente Pokemon_idPokemon se idPokemon for alterado
17  CONSTRAINT `fk_Pokemon_has_Movimento_Movimento1` -- Define uma chave estrangeira que liga Movimento_idMovimento à coluna idMovimento na tabela Movimento
18    FOREIGN KEY (`Movimento_idMovimento`)
19      REFERENCES `Pokedex`.`Movimento` (`idMovimento`)
20  ON DELETE RESTRICT                      -- Restringe a exclusão se houver associações a Pokémons
21  ON UPDATE CASCADE,                      -- Atualiza automaticamente Movimento_idMovimento se idMovimento for alterado
22  CONSTRAINT `fk_Pokemon_has_Movimento_FormadeAprender` -- Define uma chave estrangeira que liga FormadeAprender_idFormaAprender à coluna idFormaAprender na tabela FormadeAprender
23    FOREIGN KEY (`FormadeAprender_idFormaAprender`)
24      REFERENCES `Pokedex`.`FormadeAprender` (`idFormaAprender`)
25  ON DELETE RESTRICT                      -- Restringe a exclusão se houver associações a formas de aprendizado
26  ON UPDATE CASCADE,                      -- Atualiza automaticamente FormadeAprender_idFormaAprender se idFormaAprender for alterado
27 ENGINE = InnoDB;
```

```
1 -- -----
2 -- Reativa as verificações de unicidade de chave e integridade referencial
3 --
4 SET SQL_MODE=@OLD_SQL_MODE;
5 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
6 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
7
```

b) Exemplos de ALTER TABLE e DROP TABLE

```
● ● ●

1 -- Adiciona uma nova coluna `dataCaptura` à tabela `Pokemon` para armazenar a data de captura dos Pokémons.
2 -- O tipo da coluna é DATE e permite valores nulos por padrão.
3 ALTER TABLE `Pokemon`
4 ADD COLUMN `dataCaptura` DATE DEFAULT NULL;
5
6 -- Modifica a coluna `nomeTipo` na tabela `Tipos` para garantir que não permita valores nulos.
7 -- Além disso, o tamanho máximo da coluna é definido como 60 caracteres.
8 ALTER TABLE `Tipos`
9 MODIFY COLUMN `nomeTipo` VARCHAR(60) NOT NULL;
10
11 -- Adiciona uma restrição de unicidade na coluna `nome` da tabela `Movimento`.
12 -- Isso garante que não existam dois movimentos com o mesmo nome na tabela.
13 ALTER TABLE `Movimento`
14 ADD UNIQUE (`nome`);
15
16 -- Cria uma nova tabela chamada `TabelaExtra` se ela ainda não existir.
17 -- A tabela tem duas colunas: `idExtra`, um identificador único que aumenta automaticamente,
18 -- e `descricao`, que armazena uma descrição com até 100 caracteres.
19 CREATE TABLE IF NOT EXISTS `TabelaExtra` (
20   `idExtra` INT NOT NULL AUTO_INCREMENT,
21   `descricao` VARCHAR(100) NOT NULL,
22   PRIMARY KEY (`idExtra`)
23 ) ENGINE = InnoDB;
24
25 -- Remove a tabela `TabelaExtra` se ela existir.
26 -- Isso é útil para limpar a tabela antes de criar uma nova ou realizar outra operação.
27 DROP TABLE IF EXISTS `TabelaExtra`;
```

c) Inserção de dados em cada uma das tabelas

```
● ● ●  
1  -- 1. Inserindo 20 registros na tabela `Natureza`  
2  INSERT INTO Natureza (nomeNatureza)  
3  VALUES  
4  ('Hardy'),  
5  ('Lonely'),  
6  ('Brave'),  
7  ('Adamant'),  
8  ('Naughty'),  
9  ('Bold'),  
10  ('Docile'),  
11  ('Relaxed'),  
12  ('Impish'),  
13  ('Lax'),  
14  ('Timid'),  
15  ('Hasty'),  
16  ('Serious'),  
17  ('Jolly'),  
18  ('Naive'),  
19  ('Modest'),  
20  ('Mild'),  
21  ('Quiet'),  
22  ('Bashful'),  
23  ('Rash'),  
24  ('Calm'),  
25  ('Gentle'),  
26  ('Sassy'),  
27  ('Careful'),  
28  ('Quirky'),  
29  ('Timid'),  
30  ('Hasty'),  
31  ('Serious'),  
32  ('Jolly'),  
33  ('Naive');
```

```
● ○ ●
1 -- 2. Inserindo 20 registros na tabela `Pokemon`
2 INSERT INTO Pokemon (Natureza_idNatureza, numPokedex, nome, altura, peso, nivel)
3 VALUES
4 (1, 1, 'Bulbasaur', 0.7, 6.9, 5),
5 (2, 4, 'Charmander', 0.6, 8.5, 8),
6 (3, 7, 'Squirtle', 0.5, 9.0, 7),
7 (4, 25, 'Pikachu', 0.4, 6.0, 10),
8 (5, 10, 'Caterpie', 0.3, 2.9, 3),
9 (6, 16, 'Pidgey', 0.3, 1.8, 4),
10 (7, 19, 'Rattata', 0.3, 3.5, 5),
11 (8, 27, 'Sandshrew', 0.6, 12.0, 7),
12 (9, 35, 'Clefairy', 0.6, 7.5, 6),
13 (10, 39, 'Jigglypuff', 0.5, 5.5, 5),
14 (11, 50, 'Diglett', 0.2, 0.8, 5),
15 (12, 56, 'Mankey', 0.5, 28.0, 7),
16 (13, 66, 'Machop', 0.8, 19.5, 9),
17 (14, 74, 'Geodude', 0.4, 20.0, 6),
18 (15, 81, 'Magnemite', 0.3, 6.0, 4),
19 (16, 92, 'Gastly', 1.3, 0.1, 8),
20 (17, 133, 'Eevee', 0.3, 6.5, 5),
21 (18, 147, 'Dratini', 1.8, 3.3, 7),
22 (19, 150, 'Mewtwo', 2.0, 122.0, 70),
23 (20, 151, 'Mew', 0.4, 4.0, 50);
```

```
● ○ ●
1 -- 3. Inserindo 20 registros na tabela `Status`
2 INSERT INTO Status (Pokemon_idPokemon, dataHoraAtualizacao, vida, ataque, defesa, defesaEspecial, ataqueEspecial, velocidade)
3 VALUES
4 (1, '2024-01-01 12:00:00', 45, 49, 49, 65, 65, 45),
5 (2, '2024-01-01 12:00:00', 39, 52, 43, 50, 60, 65),
6 (3, '2024-01-01 12:00:00', 44, 48, 65, 50, 50, 43),
7 (4, '2024-01-01 12:00:00', 35, 55, 40, 50, 50, 90),
8 (5, '2024-01-01 12:00:00', 50, 28, 55, 25, 25, 30),
9 (6, '2024-01-01 12:00:00', 40, 45, 40, 35, 35, 56),
10 (7, '2024-01-01 12:00:00', 30, 56, 35, 25, 25, 72),
11 (8, '2024-01-01 12:00:00', 50, 75, 85, 30, 30, 40),
12 (9, '2024-01-01 12:00:00', 70, 45, 48, 60, 60, 35),
13 (10, '2024-01-01 12:00:00', 115, 45, 20, 25, 25, 20),
14 (11, '2024-01-01 12:00:00', 10, 55, 25, 35, 45, 95),
15 (12, '2024-01-01 12:00:00', 40, 80, 35, 35, 45, 70),
16 (13, '2024-01-01 12:00:00', 70, 80, 50, 35, 35, 35),
17 (14, '2024-01-01 12:00:00', 40, 80, 100, 30, 30, 20),
18 (15, '2024-01-01 12:00:00', 25, 35, 70, 95, 55, 45),
19 (16, '2024-01-01 12:00:00', 30, 35, 30, 100, 100, 80),
20 (17, '2024-01-01 12:00:00', 55, 55, 50, 45, 65, 55),
21 (18, '2024-01-01 12:00:00', 41, 64, 45, 50, 50, 50),
22 (19, '2024-01-01 12:00:00', 106, 110, 90, 90, 154, 130),
23 (20, '2024-01-01 12:00:00', 100, 100, 100, 100, 100, 100);
```



```
1 -- 4. Inserindo 18 registros na tabela `Tipos`  
2 INSERT INTO Tipos (nomeTipo)  
3 VALUES  
4 ('Fogo'),  
5 ('Água'),  
6 ('Gramá'),  
7 ('Voador'),  
8 ('Lutador'),  
9 ('Veneno'),  
10 ('Elétrico'),  
11 ('Terra'),  
12 ('Pedra'),  
13 ('Psíquico'),  
14 ('Gelo'),  
15 ('Inseto'),  
16 ('Fantasma'),  
17 ('Ferro'),  
18 ('Dragão'),  
19 ('Sombrio'),  
20 ('Fada'),  
21 ('Normal');
```



```
1 -- 5. Inserindo 20 registros na tabela `FormaDeEvoluir`
2 INSERT INTO FormaDeEvoluir (FormaEvo, nomePedra, nivelEvoluir)
3 VALUES
4 ('N', NULL, 16),
5 ('N', NULL, 32),
6 ('P', 'Pedra do Fogo', NULL),
7 ('N', NULL, 18),
8 ('P', 'Pedra do Trovão', NULL),
9 ('N', NULL, 25),
10 ('N', NULL, 20),
11 ('N', NULL, 22),
12 ('P', 'Pedra da Lua', NULL),
13 ('N', NULL, 16),
14 ('N', NULL, 30),
15 ('P', 'Pedra da Água', NULL),
16 ('N', NULL, 36),
17 ('N', NULL, 28),
18 ('N', NULL, 22),
19 ('P', 'Pedra da Folha', NULL),
20 ('N', NULL, 45),
21 ('P', 'Pedra Solar', NULL),
22 ('N', NULL, 40),
23 ('P', 'Pedra do Gelo', NULL);
```

```

1 -- 6. Inserindo 20 registros na tabela `Movimento`
2 INSERT INTO Movimento (Tipos_idTipo, nome, potencia, precisao, categoriaDoMovimento, descricao) VALUES
3 (1, 'Ember', 40, 100, 'E', 'Um ataque de fogo fraco. Pode deixar o adversário com status Burn'),
4 (1, 'Sunny Day', 0, 100, 'S', 'Causa raios de sol intensos durante 5 turnos'),
5 (2, 'Hydro Pump', 120, 80, 'E', 'Explode água em alta potência para atingir o inimigo'),
6 (2, 'Surf', 95, 100, 'E', 'Uma grande onda cai sobre o inimigo'),
7 (3, 'Sleep Powder', 0, 75, 'S', 'Espalha um pó que pode fazer o inimigo dormir. Deixa o adversário com status Sleep'),
8 (3, 'Giga Drain', 60, 100, 'E', 'Usuário recupera metade do dano que é causado'),
9 (4, 'Sky Attack', 140, 90, 'F', 'Ataca com um golpe de asas. Pode causar Flinch no adversário (interrompimento de golpe)'),
10 (4, 'Mirro Move', 0, 100, 'S', 'Usuário faz o último golpe usado pelo oponente'),
11 (5, 'Karate Chop', 50, 100, 'F', 'O inimigo é atacado com um golpe certeiro. Tem grandes chances de causar um Critical Hit'),
12 (5, 'Low Kick', 0, 100, 'F', 'Quanto mais pesado o oponente for, maior será o dano'),
13 (6, 'Poison Sting', 15, 100, 'F', 'O inimigo é esfaqueado com uma farpa tóxica. Pode deixar o adversário com status Poison'),
14 (6, 'Acid', 40, 100, 'E', 'O inimigo é pulverizado com um ácido forte que derrete a pele. Pode reduzir SPECIAL DEFENSE do adversário em um estágio'),
15 (7, 'Thunderbolt', 90, 100, 'E', 'Uma forte explosão elétrica é disparada contra o inimigo. Também pode deixar o inimigo paralisado'),
16 (7, 'Thunder', 110, 70, 'E', 'Um ataque relâmpago brutal. Pode deixar o inimigo paralisado'),
17 (8, 'Sand Attack', 0, 100, 'S', 'Muita areia é atirada na cara do inimigo, reduzindo sua precisão'),
18 (8, 'Earthquake', 100, 100, 'F', 'Desencadeia um terremoto que atinge o Pokémon inimigo'),
19 (9, 'Rock Polish', 0, 100, 'S', 'O usuário lustra seu corpo para reduzir o arrasto. Pode aumentar drasticamente a estatística de velocidade'),
20 (9, 'Rock Blast', 25, 90, 'F', 'Atira pedras duras no inimigo. Duas a cinco pedras são lançadas em rápida sucessão'),
21 (10, 'Psychic', 90, 100, 'E', 'O inimigo é atingido por uma forte força telecinética'),
22 (10, 'Confusion', 50, 100, 'E', 'O alvo é atingido por uma força telecinética fraca para causar dano. Isso também pode confundir o alvo'),
23 (11, 'Ice Beam', 90, 100, 'E', 'O inimigo é atingido por um feixe de energia gelado. Também pode congelar o sólido alvo'),
24 (11, 'Blizzard', 110, 70, 'E', 'Uma nevasca uivante é convocada para atacar o inimigo. Também pode congelar o alvo'),
25 (12, 'Pin Missile', 25, 95, 'F', 'Alfinetes afiados são disparados contra o inimigo em rápida sucessão. Eles acertaram de duas a cinco vezes seguidas'),
26 (12, 'Twineedle', 25, 100, 'F', 'O inimigo é esfaqueado duas vezes por um par de ferros. Também pode envenenar o alvo'),
27 (13, 'Shadow Ball', 80, 100, 'E', 'Lança uma bolha sombria no inimigo'),
28 (13, 'Astonish', 30, 100, 'F', 'Ataca o alvo enquanto grita de forma surpreendente'),
29 (14, 'Meteor Mash', 90, 90, 'F', 'O inimigo é atingido por um soco forte disparado como um meteoro. Também pode aumentar o ataque do usuário'),
30 (14, 'Magnet Bomb', 60, 100, 'F', 'O usuário lança uma bomba de aço que atinge o alvo. Este ataque não falhará nunca'),
31 (15, 'Draco Meteor', 130, 90, 'E', 'Os cometas são convocados do céu. O recuo do ataque reduz drasticamente o Ataque Especial do usuário'),
32 (15, 'Spacial Rend', 100, 95, 'E', 'O usuário rasga o alvo junto com o espaço ao seu redor'),
33 (16, 'Payback', 50, 100, 'F', 'Se o usuário puder usar este ataque após o ataque do inimigo, seu poder será duplicado'),
34 (16, 'Assurance', 60, 100, 'F', 'Se o inimigo já tiver sofrido algum dano no mesmo turno, o poder deste ataque é duplicado'),
35 (17, 'Fairy Lock', 0, 100, 'S', 'Ao bloquear o campo de batalha, o usuário evita que todos os Pokémon fujam durante o próximo turno'),
36 (17, 'Fairy Wind', 40, 100, 'E', 'O usuário desperta um vento mágico e atinge o alvo com ele'),
37 (18, 'Headbutt', 70, 100, 'F', 'Coloca a cabeça para fora e avança direto. Isso pode fazer o inimigo recuar'),
38 (18, 'Mega Kick', 120, 75, 'F', 'O inimigo é atacado por um chute muito forte');

```

```

1 -- 7. Inserindo 20 registros na tabela `FormaDeAprender`
2 INSERT INTO FormaDeAprender (ehTreinador, nomeTreinador, ehNivel, nivelAprender, ehDisco, nomeDisco)
3 VALUES
4 (1, 'Brock', 0, NULL, 0, NULL),
5 (1, 'Misty', 0, NULL, 0, NULL),
6 (0, NULL, 1, 'Level 16', 0, NULL),
7 (0, NULL, 1, 'Level 18', 0, NULL),
8 (0, NULL, 1, 'Level 20', 0, NULL),
9 (1, 'Erika', 0, NULL, 0, NULL),
10 (1, 'Koga', 0, NULL, 0, NULL),
11 (0, NULL, 1, 'Level 25', 0, NULL),
12 (1, 'Sabrina', 0, NULL, 0, NULL),
13 (1, 'Blaine', 0, NULL, 0, NULL),
14 (0, NULL, 1, 'Level 30', 0, NULL),
15 (0, NULL, 1, 'Level 35', 0, NULL),
16 (1, 'Giovanni', 0, NULL, 0, NULL),
17 (1, 'Blue', 0, NULL, 0, NULL),
18 (0, NULL, 1, 'Level 40', 0, NULL),
19 (1, 'Cynthia', 0, NULL, 0, NULL),
20 (0, NULL, 1, 'Level 45', 0, NULL),
21 (0, NULL, 1, 'Level 50', 0, NULL),
22 (0, NULL, 0, NULL, 1, 'TM01'),
23 (0, NULL, 0, NULL, 1, 'TM02');

```



```
1 -- 8. Inserindo 20 registros na tabela `Deve_Possuir`
2 INSERT INTO Deve_Possuir (Pokemon_idPokemon, Tipos_idTipo)
3 VALUES
4 (1, 3), (1, 5),
5 (2, 1), (2, 5),
6 (3, 2), (3, 5),
7 (4, 4), (4, 5),
8 (5, 5), (5, 6),
9 (6, 5), (6, 7),
10 (7, 6), (7, 7),
11 (8, 2), (8, 8),
12 (9, 9), (9, 10),
13 (10, 11), (10, 12),
14 (11, 13), (11, 14),
15 (12, 15), (12, 16),
16 (13, 17), (13, 18),
17 (15, 1), (15, 2),
18 (16, 3), (16, 4),
19 (17, 5), (17, 6),
20 (18, 7), (18, 10);
```

```
1 -- 9. Inserindo 20 registros na tabela `Pokemon_Aprende_Movimento_Atraves`
2 INSERT INTO Pokemon_Aprende_Movimento_Atraves (Pokemon_idPokemon, Movimento_idMovimento, FormaDeAprender_idFormaAprender)
3 VALUES
4 (1, 1, 3), (1, 2, 3),
5 (2, 4, 4), (2, 5, 4),
6 (3, 6, 5), (3, 7, 5),
7 (4, 8, 6), (4, 9, 6),
8 (5, 10, 7), (5, 11, 7),
9 (6, 12, 8), (6, 13, 8),
10 (7, 14, 9), (7, 15, 9),
11 (8, 16, 10), (8, 17, 10),
12 (9, 18, 11), (9, 19, 11),
13 (10, 20, 12), (10, 1, 12),
14 (11, 2, 13), (11, 3, 13),
15 (12, 4, 14), (12, 5, 14),
16 (13, 6, 15), (13, 7, 15),
17 (14, 8, 16), (14, 9, 16),
18 (15, 10, 17), (15, 11, 17),
19 (16, 12, 18), (16, 13, 18),
20 (17, 14, 19), (17, 15, 19),
21 (18, 16, 2), (18, 17, 1),
22 (19, 18, 1), (19, 19, 1),
23 (20, 20, 2), (20, 1, 2);
```

```
1 -- 10. Inserindo 20 registros na tabela `Evolucao`
2 INSERT INTO Evolucao (nomePokemonEvo, numPokedexEvo, Pokemon_numPokedex, FormaDeEvoluir_idFormaEvo)
3 VALUES
4 ('Ivysaur', 2, 1, 1),
5 ('Venusaur', 3, 1, 2),
6 ('Charmeleon', 4, 2, 3),
7 ('Charizard', 5, 2, 4),
8 ('Wartortle', 6, 3, 5),
9 ('Blastoise', 7, 3, 6),
10 ('Pidgeotto', 8, 4, 7),
11 ('Pidgeot', 9, 4, 8),
12 ('Raticate', 10, 5, 9),
13 ('Sandslash', 11, 7, 10),
14 ('Nidorina', 12, 6, 11),
15 ('Nidoqueen', 13, 6, 12),
16 ('Nidorino', 14, 13, 13),
17 ('Nidoking', 15, 13, 14),
18 ('Clefairy', 16, 15, 15),
19 ('Clefable', 17, 15, 16),
20 ('Vulpix', 18, 17, 17),
21 ('Ninetales', 19, 17, 18),
22 ('Machoke', 20, 18, 19),
23 ('Machamp', 21, 18, 20);
```

d) Modificação de dados e uso do UPDATE aninhado

```
1 -- Atualiza o nome da natureza para 'Brava' na tabela `Natureza` onde o `idNatureza` é igual a 1.
2 UPDATE Natureza
3 SET nomeNatureza = 'Brava'
4 WHERE idNatureza = 1;
5
6 -- Altera o nome do Pokémon para 'Pikachu' e seu nível para 25 na tabela `Pokemon`
7 -- onde o `idPokemon` é igual a 2.
8 UPDATE Pokemon
9 SET nome = 'Pikachu', nivel = 25
10 WHERE idPokemon = 2;
11
12 -- Define a potência como 90 e a precisão como 100 no movimento da tabela `Movimento`
13 -- onde o `idMovimento` é igual a 5.
14 UPDATE Movimento
15 SET potencia = 90, precisao = 100
16 WHERE idMovimento = 5;
17
18 -- Atualiza o nível necessário para evoluir para 36 na tabela `FormaDeEvoluir`
19 -- onde o `idFormaEvo` é igual a 3.
20 UPDATE FormaDeEvoluir
21 SET nivelEvoluir = 36
22 WHERE idFormaEvo = 3;
23
24 -- Atualiza a vida para 120 e o ataque para 80 na tabela `Status` para o Pokémon
25 -- com `idPokemon` igual a 4 e onde a data e hora de atualização é '2024-08-20 10:00:00'.
26 UPDATE Status
27 SET vida = 120, ataque = 80
28 WHERE Pokemon_idPokemon = 4 AND dataHoraAtualizacao = '2024-08-20 10:00:00';
29
30 -- Atualiza o nível do Pokémon para 50 e o nome do treinador para 'Ash Ketchum'.
31 -- A atualização é realizada para Pokémon com vida superior a 100 e que estão sendo treinados por um treinador.
32 UPDATE Pokemon p
33 JOIN Status s ON p.idPokemon = s.Pokemon_idPokemon
34 JOIN Pokemon_Aprende_Movimento_Atraves pam ON p.idPokemon = pam.Pokemon_idPokemon
35 JOIN FormaDeAprender fda ON pam.FormaDeAprender_idFormaAprender = fda.idFormaAprender
36 SET p.nivel = 50, fda.nomeTreinador = 'Ash Ketchum'
37 WHERE s.vida > 100 AND fda.ehTreinador = 1;
```

e) Exemplos de exclusão em 5 tabelas. Uso do DELETE

```
● ● ●

1 -- Exclui a natureza da tabela `Natureza` onde o `idNatureza` é igual a 3.
2 DELETE FROM Natureza
3 WHERE idNatureza = 3;
4
5 -- Exclui o Pokémon da tabela `Pokemon` onde o `idPokemon` é igual a 5.
6 DELETE FROM Pokemon
7 WHERE idPokemon = 5;
8
9 -- Remove o movimento da tabela `Movimento` onde o `idMovimento` é igual a 10.
10 DELETE FROM Movimento
11 WHERE idMovimento = 10;
12
13 -- Exclui a forma de evolução da tabela `FormaDeEvoluir` onde o `idFormaEvo` é igual a 2.
14 DELETE FROM FormaDeEvoluir
15 WHERE idFormaEvo = 2;
16
17 -- Exclui o registro de status do Pokémon na tabela `Status` com `idPokemon` igual a 7
18 -- e a data e hora de atualização específica '2024-08-20 10:00:00'.
19 DELETE FROM Status
20 WHERE Pokemon_idPokemon = 7 AND dataHoraAtualizacao = '2024-08-20 10:00:00';
21
22 -- Exclui todos os registros de status associados ao Pokémon com `idPokemon` igual a 8
23 -- e depois remove o Pokémon da tabela `Pokemon`.
24 DELETE s, p
25 FROM Status s
26 JOIN Pokemon p ON s.Pokemon_idPokemon = p.idPokemon
27 WHERE p.idPokemon = 8;
```

f) Consultas

```
1 -- 1. Recupera o nome e a altura de todos os Pokémons.
2 SELECT nome, altura
3 FROM Pokemon;
4 -- Essa consulta retorna os nomes e as alturas de todos os Pokémons registrados na tabela `Pokemon`.
5
6 -- 2. Recupera o nome e o peso de todos os Pokémons que pesam mais de 100 kg.
7 SELECT nome, peso
8 FROM Pokemon
9 WHERE peso > 100;
10 -- Filtra e exibe os Pokémons cujo peso é superior a 100 kg.
11
12 -- 3. Recupera todos os movimentos ordenados por precisão, do mais alto para o mais baixo.
13 SELECT nome, precisao
14 FROM Movimento
15 ORDER BY precisao DESC;
16 -- Lista os movimentos ordenados pela precisão de forma decrescente, exibindo primeiro os mais precisos.
17
18 -- 4. Recupera a média de altura dos Pokémons agrupados por Natureza.
19 SELECT n.nomeNatureza, AVG(p.altura) AS mediaAltura
20 FROM Pokemon p
21 JOIN Natureza n ON p.Natureza_idNatureza = n.idNatureza
22 GROUP BY n.nomeNatureza;
23 -- Calcula e exibe a altura média dos Pokémons, agrupados por sua natureza.
24
25 -- 5. Recupera os nomes e o total de Pokémons agrupados por Natureza, mas apenas para naturezas que têm mais de 5 Pokémons.
26 SELECT n.nomeNatureza, COUNT(p.idPokemon) AS totalPokemons
27 FROM Pokemon p
28 JOIN Natureza n ON p.Natureza_idNatureza = n.idNatureza
29 GROUP BY n.nomeNatureza
30 HAVING COUNT(p.idPokemon) > 5;
31 -- Agrupa os Pokémons por Natureza e mostra apenas as naturezas que possuem mais de 5 Pokémons.
32
33 -- 6. Recupera os nomes dos Pokémons e seus respectivos movimentos, incluindo apenas aqueles com precisão maior que 80.
34 SELECT p.nome AS nomePokemon, m.nome AS nomeMovimento
35 FROM Pokemon p
36 JOIN Pokemon_Aprende_Movimento_Atraves pam ON p.idPokemon = pam.Pokemon_idPokemon
37 JOIN Movimento m ON pam.Movimento_idMovimento = m.idMovimento
38 WHERE m.precisao > 80;
39 -- Lista os nomes dos Pokémons e os movimentos que eles aprendem, mas somente aqueles movimentos cuja precisão é maior que 80.
40
41 -- 7. Recupera todos os tipos de Pokémons que ainda não possuem nenhum movimento associado.
42 SELECT t.nomeTipo
43 FROM Tipos t
44 LEFT JOIN Movimento m ON t.idTipo = m.Tipos_idTipo
45 WHERE m.idMovimento IS NULL;
46 -- Exibe os tipos de Pokémons que não têm nenhum movimento associado.
47
48 -- 8. Recupera todos os Pokémons que estão entre os níveis 10 e 20.
49 SELECT nome, nivel
50 FROM Pokemon
51 WHERE nivel BETWEEN 10 AND 20;
52 -- Lista os Pokémons que estão no intervalo de nível entre 10 e 20, inclusive.
```

```
1
2 -- 9. Recupera o nome dos Pokémons que possuem o movimento com o ID 5 ou 10.
3 SELECT p.nome
4 FROM Pokemon p
5 JOIN Pokemon_Aprende_Movimento_Atraves pam ON p.idPokemon = pam.Pokemon_idPokemon
6 WHERE pam.Movimento_idMovimento IN (5, 10);
7 -- Exibe o nome dos Pokémons que aprenderam os movimentos com IDs 5 ou 10.
8
9 -- 10. Recupera o nome dos Pokémons que possuem nível superior ao de todos os Pokémons com natureza "Brava".
10 SELECT p.nome
11 FROM Pokemon p
12 WHERE p.nivel > ALL (
13     SELECT p2.nivel
14     FROM Pokemon p2
15     JOIN Natureza n ON p2.Natureza_idNatureza = n.idNatureza
16     WHERE n.nomeNatureza = 'Brava'
17 );
18 -- Lista os Pokémons cujo nível é superior ao nível de todos os Pokémons que têm natureza "Brava".
19
20 -- 11. Recupera os nomes dos Pokémons que possuem pelo menos um movimento de tipo 'Fogo'.
21 SELECT p.nome
22 FROM Pokemon p
23 WHERE EXISTS (
24     SELECT 1
25     FROM Pokemon_Aprende_Movimento_Atraves pam
26     JOIN Movimento m ON pam.Movimento_idMovimento = m.idMovimento
27     JOIN Tipos t ON m.Tipos_idTipo = t.idTipo
28     WHERE pam.Pokemon_idPokemon = p.idPokemon
29     AND t.nomeTipo = 'Fogo'
30 );
31 -- Recupera o nome dos Pokémons que têm pelo menos um movimento do tipo 'Fogo'. A subconsulta verifica a existência desse movimento para cada Pokémon.
32
33 -- 12. Recupera o nome e o total de movimentos de cada Pokémon que tem mais de 3 movimentos.
34 SELECT p.nome, COUNT(m.idMovimento) AS totalMovimentos
35 FROM Pokemon p
36 JOIN Pokemon_Aprende_Movimento_Atraves pam ON p.idPokemon = pam.Pokemon_idPokemon
37 JOIN Movimento m ON pam.Movimento_idMovimento = m.idMovimento
38 GROUP BY p.nome
39 HAVING COUNT(m.idMovimento) > 3;
40 -- Exibe o nome de cada Pokémon e o número total de movimentos que ele aprende, mas apenas para aqueles que têm mais de 3 movimentos.
41
```

g) Views

h) Criação de Usuários

```
● ● ●

1 -- 1. Criação de Usuários
2
3 -- Criação do usuário 'usuario_admin' com senha 'senha_admin'
4 -- Esse usuário terá permissões administrativas no banco de dados
5 CREATE USER 'usuario_admin'@'localhost' IDENTIFIED BY 'senha_admin';
6
7 -- Criação do usuário 'usuario_leitor' com senha 'senha_leitor'
8 -- Esse usuário terá permissões limitadas, apenas para leitura
9 CREATE USER 'usuario_leitor'@'localhost' IDENTIFIED BY 'senha_leitor';
10
11 -- 2. Concessão de Permissões
12
13 -- Concede todas as permissões no banco de dados 'Pokedex' ao usuário 'usuario_admin'
14 -- Isso inclui a capacidade de SELECT, INSERT, UPDATE, DELETE, e outras operações administrativas
15 GRANT ALL PRIVILEGES ON Pokedex.* TO 'usuario_admin'@'localhost';
16
17 -- Concede permissões de seleção (leitura) no banco de dados 'Pokedex' ao usuário 'usuario_leitor'
18 -- O usuário pode realizar operações de leitura (SELECT) nas tabelas
19 GRANT SELECT ON Pokedex.* TO 'usuario_leitor'@'localhost';
20
21 -- 3. Revogação de Permissões
22
23 -- Revoga todas as permissões do banco de dados 'Pokedex' do usuário 'usuario_admin'
24 -- Isso remove todas as permissões administrativas anteriormente concedidas
25 REVOKE ALL PRIVILEGES ON Pokedex.* FROM 'usuario_admin'@'localhost';
26
27 -- Revoga permissões de seleção (leitura) no banco de dados 'Pokedex' do usuário 'usuario_leitor'
28 -- O usuário não poderá mais realizar operações de leitura nas tabelas
29 REVOKE SELECT ON Pokedex.* FROM 'usuario_leitor'@'localhost';
30
31 -- 4. Aplicar as Mudanças
32
33 -- Atualiza os privilégios para garantir que as mudanças sejam aplicadas
34 -- Isso faz com que todas as concessões e revogações de permissões se tornem efetivas
35 FLUSH PRIVILEGES;
```

i)Procedimentos/Funções

```
● ● ●

1  DELIMITER //
2
3  -- 1. Criação do procedimento `sp_InserirPokemon`
4  CREATE PROCEDURE sp_InserirPokemon(
5      IN p_Nome VARCHAR(40),          -- Nome do Pokémon a ser inserido
6      IN p_Natureza_idNatureza INT,   -- ID da natureza do Pokémon
7      IN p_numPokedex INT,           -- Número da Pokédex do Pokémon
8      IN p_altura FLOAT,            -- Altura do Pokémon
9      IN p_peso FLOAT,              -- Peso do Pokémon
10     IN p_nivel INT               -- Nível do Pokémon
11  )
12 BEGIN
13     -- Inserção de dados na tabela `Pokemon`
14     INSERT INTO Pokemon (nome, Natureza_idNatureza, numPokedex, altura, peso, nivel)
15     VALUES (p_Nome, p_Natureza_idNatureza, p_numPokedex, p_altura, p_peso, p_nivel);
16
17     -- Verificação da inserção com SELECT
18     IF ROW_COUNT() > 0 THEN
19         SELECT 'Inserção bem-sucedida!' AS Resultado; -- Mensagem indicando sucesso
20     ELSE
21         SELECT 'Falha na inserção.' AS Resultado;       -- Mensagem indicando falha
22     END IF;
23 END //
24
25 DELIMITER ;
26
27 -- Executa o procedimento `sp_InserirPokemon` para adicionar um novo Pokémon
28 CALL sp_InserirPokemon('Pikachu', 1, 25, 0.4, 6.0, 5);
29
30 DELIMITER //
31
32 -- 2. Criação da função `fn_CalcularMediaAltura`
33 CREATE FUNCTION fn_CalcularMediaAltura(p_numPokedex INT)
34 RETURNS FLOAT                  -- Retorna um valor do tipo FLOAT
35 DETERMINISTIC                 -- A função sempre retorna o mesmo resultado para os mesmos parâmetros
36 BEGIN
37     DECLARE v_mediaAltura FLOAT;    -- Declara uma variável para armazenar a média de altura
38
39     -- Cálculo da média de altura
40     SELECT AVG(altura) INTO v_mediaAltura
41     FROM Pokemon
42     WHERE numPokedex = p_numPokedex;
43
44     RETURN v_mediaAltura;          -- Retorna a média de altura calculada
45 END //
```

```
1  DELIMITER ;
2
3 -- Executa a função `fn_CalcularMediaAltura` para obter a média de altura do Pokémons com número 25
4 SELECT fn_CalcularMediaAltura(25) AS MediaAltura;
5
6 DELIMITER //
7
8 -- 3. Criação do procedimento `sp_AtualizarStatusPokemon`
9 CREATE PROCEDURE sp_AtualizarStatusPokemon(
10    IN p_Pokemon_idPokemon INT,          -- ID do Pokémons a ser atualizado
11    IN p_dataHoraAtualizacao DATETIME,   -- Data e hora da atualização
12    IN p_vida INT,                      -- Valor da vida do Pokémons
13    IN p_ataque INT,                   -- Valor do ataque do Pokémons
14    IN p_defesa INT,                  -- Valor da defesa do Pokémons
15    IN p_defesaEspecial INT,           -- Valor da defesa especial do Pokémons
16    IN p_ataqueEspecial INT,           -- Valor do ataque especial do Pokémons
17    IN p_velocidade INT               -- Valor da velocidade do Pokémons
18 )
19 BEGIN
20     -- Atualiza o status do Pokémons na tabela `Status`
21     UPDATE Status
22     SET vida = p_vida,
23         ataque = p_ataque,
24         defesa = p_defesa,
25         defesaEspecial = p_defesaEspecial,
26         ataqueEspecial = p_ataqueEspecial,
27         velocidade = p_velocidade
28     WHERE Pokemon_idPokemon = p_Pokemon_idPokemon
29     AND dataHoraAtualizacao = p_dataHoraAtualizacao;
30
31     -- Verificação da atualização com SELECT
32     IF ROW_COUNT() > 0 THEN
33         SELECT 'Status atualizado com sucesso!' AS Resultado; -- Mensagem indicando sucesso
34     ELSE
35         SELECT 'Nenhuma atualização feita.' AS Resultado;      -- Mensagem indicando que nenhuma linha foi atualizada
36     END IF;
37 END //
38
39 DELIMITER ;
40
41 -- Executa o procedimento `sp_AtualizarStatusPokemon` para atualizar o status do Pokémons com ID 1
42 CALL sp_AtualizarStatusPokemon(1, '2024-01-01 12:00:00', 100, 50, 50, 40, 60, 70);
43
```

j) Triggers

```
● ● ●

1 -- 1. Tabela de log para registrar inserções na tabela `Pokemon`
2 CREATE TABLE IF NOT EXISTS Pokemon_Insercao_Log (
3     id INT NOT NULL AUTO_INCREMENT, -- Identificador único para cada registro de log
4     Pokemon_idPokemon INT NOT NULL, -- ID do Pokémon inserido
5     Nome VARCHAR(40) NOT NULL, -- Nome do Pokémon inserido
6     Data_Insercao DATETIME NOT NULL, -- Data e hora da inserção
7     PRIMARY KEY (id) -- Chave primária que garante a unicidade dos registros
8 );
9
10 -- 2. Tabela de log para registrar atualizações de status na tabela `Status`
11 CREATE TABLE IF NOT EXISTS Status_Atualizacao_Log (
12     id INT NOT NULL AUTO_INCREMENT, -- Identificador único para cada registro de log
13     Pokemon_idPokemon INT NOT NULL, -- ID do Pokémon cujo status foi atualizado
14     Data_Hora_Atualizacao DATETIME NOT NULL, -- Data e hora da atualização
15     PRIMARY KEY (id) -- Chave primária que garante a unicidade dos registros
16 );
17
18 -- 3. Tabela de log para registrar exclusões na tabela `Pokemon`
19 CREATE TABLE IF NOT EXISTS Pokemon_Exclusao_Log (
20     id INT NOT NULL AUTO_INCREMENT, -- Identificador único para cada registro de log
21     Pokemon_idPokemon INT NOT NULL, -- ID do Pokémon excluído
22     Nome VARCHAR(40) NOT NULL, -- Nome do Pokémon excluído
23     Data_Exclusao DATETIME NOT NULL, -- Data e hora da exclusão
24     PRIMARY KEY (id) -- Chave primária que garante a unicidade dos registros
25 );
26
27 -- Agora vamos criar os triggers.
28
29 DELIMITER //
30
31 -- 1. Trigger para registrar inserções na tabela `Pokemon`
32 CREATE TRIGGER trg_AposInserirPokemon
33 AFTER INSERT ON Pokemon
34 FOR EACH ROW
35 BEGIN
36     -- Este trigger é acionado após a inserção de um novo Pokémon na tabela `Pokemon`.
37     -- Ele insere um registro na tabela `Pokemon_Insercao_Log` para manter um log da inserção.
38     INSERT INTO Pokemon_Insercao_Log (Pokemon_idPokemon, Nome, Data_Insercao)
39     VALUES (NEW.idPokemon, NEW.nome, NOW()); -- `NEW` refere-se à nova linha inserida na tabela `Pokemon`
40 END //
```

```
1 -- 2. Trigger para registrar atualizações de status na tabela 'Status'
2 CREATE TRIGGER trg_AposAtualizarStatus
3 AFTER UPDATE ON Status
4 FOR EACH ROW
5 BEGIN
6     -- Este trigger é acionado após uma atualização na tabela 'Status'.
7     -- Ele insere um registro na tabela 'Status_Atualizacao_Log' para manter um log da atualização.
8     INSERT INTO Status_Atualizacao_Log (Pokemon_idPokemon, Data_Hora_Atualizacao)
9     VALUES (NEW.Pokemon_idPokemon, NOW()); -- 'NEW' refere-se à linha atualizada na tabela 'Status'
10 END //
11
12 -- 3. Trigger para registrar exclusões na tabela 'Pokemon'
13 CREATE TRIGGER trg_AposExcluirPokemon
14 AFTER DELETE ON Pokemon
15 FOR EACH ROW
16 BEGIN
17     -- Este trigger é acionado após a exclusão de um Pokémon da tabela 'Pokemon'.
18     -- Ele insere um registro na tabela 'Pokemon_Exclusao_Log' para manter um log da exclusão.
19     INSERT INTO Pokemon_Exclusao_Log (Pokemon_idPokemon, Nome, Data_Exclusao)
20     VALUES (OLD.idPokemon, OLD.nome, NOW()); -- 'OLD' refere-se à linha excluída da tabela 'Pokemon'
21 END //
22
23 DELIMITER ;
24
25 -- Agora, vamos mostrar como esses triggers são acionados com exemplos de inserção, atualização e exclusão.
26
27 -- 1. Exemplo de inserção que aciona o trigger 'trg_AposInserirPokemon'
28 INSERT INTO Pokemon (nome, Natureza_idNatureza, numPokedex, altura, peso, nivel)
29 VALUES ('Bulbasaur', 1, 1, 0.7, 6.9, 5); -- Isso acionará o trigger para registrar a inserção do Pokémon Bulbasaur.
30
31 -- 2. Exemplo de atualização que aciona o trigger 'trg_AposAtualizarStatus'
32 UPDATE Status
33 SET vida = 95
34 WHERE Pokemon_idPokemon = 1 AND dataHoraAtualizacao = '2024-08-20 10:00:00'; -- Isso acionará o trigger para registrar a atualização do status do Pokémon com ID 1.
35
36 -- 3. Exemplo de exclusão que aciona o trigger 'trg_AposExcluirPokemon'
37 DELETE FROM Pokemon
38 WHERE idPokemon = 1; -- Isso acionará o trigger para registrar a exclusão do Pokémon com ID 1.
39
```

Repositório com os Códigos para teste:

<https://github.com/depaulatiago/trabalhosISBD>