

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CIÊNCIA DA COMPUTAÇÃO

ARIEL SADETSKY: 793240
DANILLO SANTIAGO OLIVEIRA DA CONCEIÇÃO: 801487
VANDERLEI GUILHERME ANDRADE DE ASSIS: 802162
GRUPO 12

Banco de Dados

Banco de Dados de Categorização e Catalogação de Pokémon

Tema – Pokédex Fase 2 - Entrega: 28/02/23

Professora responsável: Sahudy Montenegro González

SOROCABA
2023

Sumário

1. Descrição do problema, minimundo e consultas a serem realizadas

1.1 Consultas.

2. Modelagem do problema

3. Metadados

4. Modelo Relacional

5. Projeto físico de BD

6. Consultas

7.Triggers - view

8. Considerações Finais

1. Descrição do problema, minimundo e consultas a serem realizadas

Pokémon é uma franquia de jogos, desenho, quadrinhos e brinquedos que surgiu em 1997, se tornando um sucesso monstruoso. Ao longo dos anos, Pokémon criou uma legião de fãs apaixonados, tornando assim a propriedade intelectual mais valiosa e lucrativa do mundo. Mediante ao seu grande sucesso e influência na cultura pop atual, e a rápida expansão do número de criaturas, alcançando a casa dos 4 dígitos em 2022, se torna mister a criação de uma ferramenta de categorização e catalogação dos monstros digitais, para que, seu impacto seja compreendido por todos aqueles que sejam desconhecedores em relação ao tema a dimensão deste mundo. Dessa forma, a criação de uma pokédex funcional - enciclopédia pokémon- se torna um meio para esse fim, uma vez que, a sua idealização na ficção tem por funcionalidade armazenar em um banco de dados todos os monstros por suas características, além de informações relacionadas do mundo com o treinador, funcionando assim, como um registro universal daquele mundo. Cada pokémon possui um nome, um id code que é um identificador único, região de origem e um nível. Dois pokémon diferentes possuem um relacionamento ecológico entre si, além disso, todo monstro possui um tipo. Todo treinador possui pelo menos 1 e no máximo 6 pokémon.

Um treinador possui nome, número de pokémon, número de insígnias. Cada treinador é orientado por um professor, um professor possui um nome, uma especialidade em um determinado tipo ou pokémon, e vive em uma determinada região. Cada professor é subordinado a uma liga. Uma liga tem um nome e pertence a uma região exclusivamente. Ela certifica a competência dos ginásios, além de organizar torneios regionais. Para um treinador participar de um torneio regional, ele precisa de insígnias que são as recompensas por vencerem um ginásio. Cada ginásio possui um líder (que é um treinador) e outros treinadores que podem participar do ginásio. Também possui uma insígnia, uma localidade, e um ID. Os treinadores participam de torneios que possuem um nome e pertencem a uma região. Para participar de um torneio, um treinador precisa possuir pelo menos 4 insígnias.

*** O plural de “Pokémon” é a própria palavra “Pokémon”.**

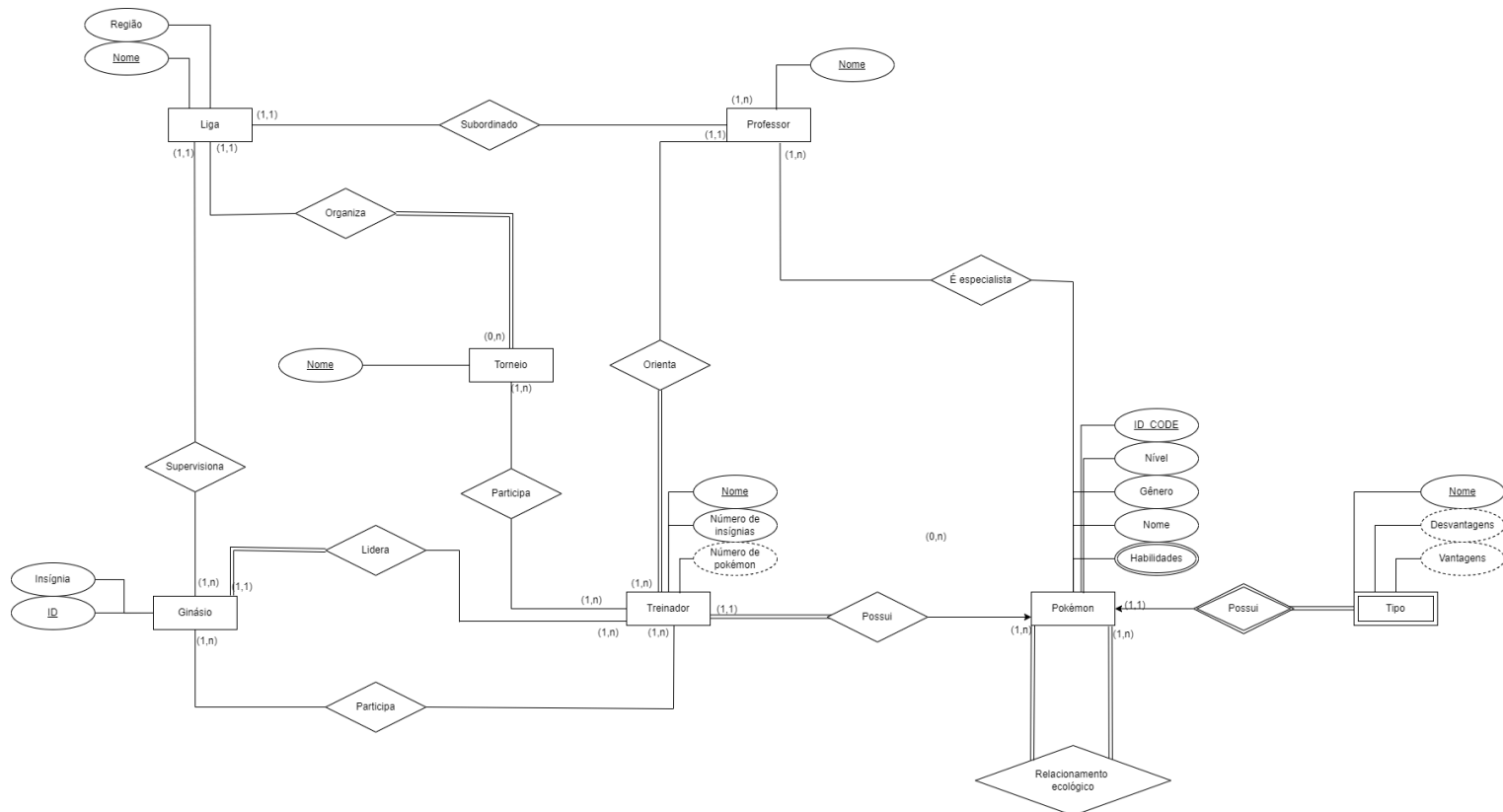
1.1 Consultas.

Tabela 1 - Consultas/Responsável

Consultas	Responsável
“Quais pokémon um treinador possui?”	Ariel Sadetsky
“Quantas insígnias um treinador já possui?”	
“Qual relação ecológica uma espécie de pokémon tem com outros?”	
“Se determinado treinador pode participar de um torneio”	Danillo Santiago Oliveira da Conceição
“Qual o pokémon de menor nível de um determinado treinador?”	
“Quais e quantas vezes um treinador perdeu para um líder de ginásio ?”	
“Qual a média dos níveis dos pokémon de um determinado treinador?”	Vanderlei Guilherme Andrade de Assis
“Quantas vantagens um pokémon tem?”	
“Quantos ginásios têm em determinada região”	

Consultas implementadas pelo grupo ao final da última entrega do projeto

2. Modelagem do problema



MER que descreve o problema apresentado

Para o mapeamento do problema em um Diagrama Entidade Relacionamento (MER), tomamos o minimundo como base. Toda descrição dada gira em torno dos Pokémon e seus treinadores que possuem suas características específicas, tornando-se entidades no diagrama apresentado acima. A Liga que está presente na descrição tem um papel regulador com relação a outras organizações e por isso está no MER como a única entidade que não se relaciona diretamente com Pokémon ou Treinador. Os Torneios se encaixam no diagrama sendo organizados pela Liga e permitindo a participação de treinadores, sendo uma outra organização baseada em batalhas, assim como os ginásios, que além de permitirem a participação de treinadores, possuem um líder em sua estrutura que também é um treinador. Com isso, a entidade Ginásio tem dois relacionamentos com treinador: o primeiro indicando que cada um deles é liderado por um ou mais treinadores e o outro que trata da participação dos treinadores que buscam a insígnia do ginásio. A entidade Professor aparece no modelo se relacionando com 3 entidades: orientando um ou mais treinadores, podendo ser especialista em um ou mais Pokémon e sendo subordinado a uma Liga tendo um papel de apoio aos treinadores nesse cenário. Pokémon também têm um auto relacionamento já que, de acordo com a descrição, eles exercem relações ecológicas entre si. Por fim, Tipo aparece como uma entidade fraca, já que não é possível haver um tipo de Pokémon caso a criatura não exista.

* O atributo nome da Entidade Pokémon se refere ao nome da espécie

3. Metadados

Tabela 1 - Metadados Pokémon

Tipo-Entidade	Atributo	Tipo	Restrição
Pokémon	Gênero	Monovalorado	Opcional (3 valores possíveis: macho, fêmea e sem gênero)
	Habilidades	Multivalorado	Obrigatório (precisa ter 1 ou mais habilidades)
	ID_code	Identificador	Obrigatório (um número maior ou igual a 1)
	Nível	Monovalorado	Obrigatório (número maior ou igual a 0)
	Nome	Monovalorado	Obrigatório (String)

Tabela 2 - Metadados Tipo

Tipo-Entidade	Atributo	Tipo	Restrição
Tipo	Nome	Identificador	Obrigatório (String)

Tabela 3 - Metadados Treinador

Tipo-Entidade	Atributo	Tipo	Restrição
Treinador	Nome	Identificador	Obrigatório (String)
	Número de pokémon	Monovalorado	Obrigatório (número maior ou igual a 1)
	Número de insígnias	Monovalorado	Obrigatório (número maior ou igual a 0)

Tabela 4 - Metadados Professor

Tipo-Entidade	Atributo	Tipo	Restrição
Professor	Nome	Identificador	Obrigatório (String)

Tabela 5 - Metadados Ginásio

Tipo-Entidade	Atributo	Tipo	Restrição
Ginásio	ID	Identificador	Obrigatório (Número maior ou igual a 1)
	Insígnia	Monovalorado	Obrigatório (String)

Tabela 6 - Metadados Liga

Tipo-Entidade	Atributo	Tipo	Restrição
Liga	Nome	Identificador	Obrigatório (String)
	Região	Monovalorado	Obrigatório (String)

Tabela 7 - Metadados Torneio

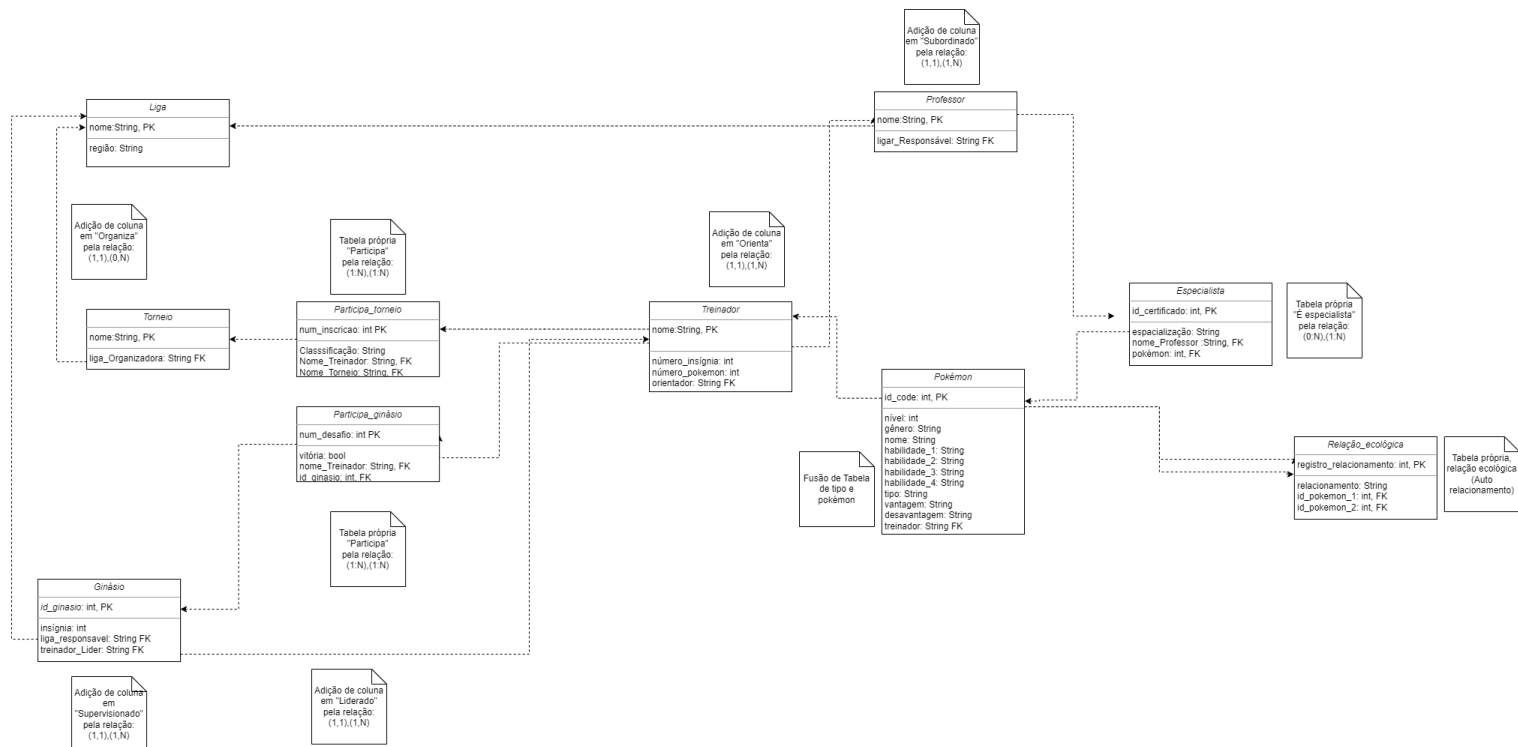
Tipo-Entidade	Atributo	Tipo	Restrição
Torneio	Nome	Identificador	Obrigatório (String)

Descrição mais detalhada para cada atributo de cada Tipo-Entidade do MER

A tabela de metadados descreve melhor cada Tipo-Entidade presente no Modelo Entidade-Relacionamento. Nele podem ser observadas informações como os atributos pertencentes a determinada entidade e informações sobre ele como seu tipo: ele pode ser um atributo identificador, monovalorado ou multivalorado. Também são dadas as restrições que atuam sobre este atributo. Dentre elas, é dito se o atributo é obrigatório ou opcional (pode assumir valores nulos ou não) e qual o domínio de cada atributo (qual tipo de dado deverá ser usado para armazenar o valor atribuído a cada atributo e quais serão esses possíveis valores).

4. Modelo Relacional

Link para o GitHub com o modelo relacional em maior qualidade: [Modelo](#)



MR mapeado a partir do MER com certas normalizações

O Modelo Relacional apresentado acima contém 10 tabelas:

- A tabela Liga cuja chave primária é uma string “nome” e contém um atributo “região” cujo domínio também é String e armazena o nome de uma região em que determinada liga se encontra.
- A tabela Professor, cuja chave primária é uma string “nome” e contém uma chave estrangeira (vinda do atributo “nome” da tabela Liga) chamada de “Liga_Responsável”, já que cada professor deve estar associado a uma liga.
- A tabela Torneio, cuja chave primária é uma string “nome” e contém uma chave estrangeira (vinda do atributo “nome” da tabela Liga) chamada de “liga_Organizadora”, já que cada torneio é organizado por uma liga.
- A tabela Treinador, cuja chave primária é uma string “nome”, além de possuir dois atributos com números inteiros: “número_insígnia” e “número_pokemon” que representam a quantidade de insígnias já obtidas por um treinador até o momento (que equivale a quantidade de ginásios que já foram vencidos por esse treinador) e a quantidade de pokémon que ele já possui respectivamente. Também há uma chave estrangeira (vinda do atributo “nome” da tabela Professor) chamada de “orientador”, pois cada treinador é orientado por um professor.
- A tabela Pokémon, cuja chave primária é um índice inteiro “id_code” além de ter um atributo inteiro com o seu nível (uma característica dos pokémon que indica sua experiência), uma string com seu gênero (que pode ser masculino, feminino ou pode não existir), uma string nome (que indica o nome de determinada espécie de pokémon), 4 strings de habilidades

listadas de 1 a 4 (indicam as ações que um pokémon pode tomar em uma batalha), uma string “tipo” que determina sua vantagem e desvantagem (outras duas strings que também são atributos de Pokémon) e, por fim, uma chave estrangeira (vinda do atributo “nome” da tabela Treinador) chamada de “treinador”, pois cada pokémon é treinado por um treinador.

- A tabela Especialista, cuja PK é um índice inteiro “id_certificado”, além de ter um atributo do tipo string que descreve qual é a especialização obtida pelo Professor (chamada de “especialização”) e duas chaves estrangeiras: a primeira vinda do atributo “nome” da tabela Professor, chamada de “nome_Professor” e a segunda vinda do atributo “id_code” da tabela Pokémon, chamada de “pokémon”. Devido a um professor se especializar em um pokémon.
- A tabela “Relação_ecológica” cuja chave primária é um índice inteiro “registro_relacionamento”, além de ter um atributo do tipo string “relacionamento” que indica o tipo da relação exercida entre dois pokémon e duas chaves estrangeiras vindas do atributo “nome” da tabela Pokémon, que indicam os dois atores de determinada relação ecológica.
- A tabela Ginásio, cuja chave primária é um índice inteiro “id_ginasio”, além de ter um atributo inteiro “insígnia” que representa um número identificador e duas chaves estrangeiras: a primeira vinda do atributo “nome” da tabela Liga, chamada de “liga_responsavel”, que indica qual é a liga que regula determinado ginásio e a segunda vinda do atributo “nome” da tabela Treinador, chamada de “Treinador_Lider”, que indica qual é o treinador que assumiu a função de líder de determinado ginásio.
- A tabela Participa_torneio cuja PK é um índice inteiro “num_inscricao”, além de ter um atributo do tipo string “classificação” que indica a posição final que um treinador alcançou em um determinado torneio e duas chaves estrangeiras: uma vinda do atributo “nome” da tabela Treinador e a outra vinda do atributo “nome” da tabela Torneio chamadas respectivamente de “Nome_treinador” e “Nome_Torneio” indicando qual treinador que está a participar de qual torneio.
- A tabela Participa_ginásio cuja PK é um índice inteiro “num_desafio”, além de ter um atributo do tipo booleano “vitória” que indica se determinado treinador já venceu certo ginásio e duas chaves estrangeiras: Uma chamada de “nome_Treinador” vinda do atributo “nome” da tabela Treinador e a outra chamada de “id_ginasio” vinda do atributo “id_ginasio” da tabela Ginásio, indicando qual treinador que está a desafiar determinado ginásio.

Utilizamos como base, as regras de mapeamento. Assim optamos por fundir as entidades “Pokémon” e “tipo”, como os demais casos explicados no próprio modelo, reduzindo o número de chaves primárias e evitando junções. Para os relacionamentos cuja cardinalidade era de 1 para N, fizemos a adição de coluna. O que evita possíveis campos opcionais e diminui o número de chaves primárias comparada a solução de se criar novas tabelas. Para o relacionamento “Especialista”, entre “Professor” e “Pokémon”, foi feita uma tabela própria devido à cardinalidade ser de N para M. Para eliminar o atributo multivalorado “Habilidade” decidimos separar ele em 4 campos, pois normalmente os Pokémon possuem esta quantidade padrão de habilidades.

O modelo atende a forma normal 3 (3FN) pois:

No caso o modelo atende a 1FN, pois retiramos todos os atributos multivalorados, criando uma tabela própria para ela ou criando campos equivalentes, como feito em habilidade, que trocamos esse atributo multivalorado por 4 campos habilidade _(n), na tabela “Pokémon”.

O modelo também atende a 2FN:

Por não haver dependência funcional parcial em nenhuma relação e por já atendermos a 1FN.

O modelo atende a 3FN:

Por não existir nenhuma dependência funcional transitiva, e, em retrospectiva, atendemos à 2FN.

5. Projeto físico de BD

Os scripts de criação do banco de dados se encontram anexados no arquivo ddl.sql e os scripts de inserção de dados no banco de dados se encontram anexados no arquivo dados.sql, as consultas do banco de dados estão no arquivo consulta.sql, e o código da trigger e view se encontram no arquivo trigger_view.sql

Cláusulas utilizadas na DDL

- **NOT NULL:**

- nome (Pokémon)
- tipo (Pokemon)
- vantagem (Pokemon)
- desvantagem(Pokemon)
- região (liga)
- liga responsável (Professor)
- liga responsável (Torneio)
- liga responsável (Ginásio)
- insígnia (Ginásio)
- treinador líder (Ginásio)
- classificação (Participa Torneio)
- nome participante (Participa torneio)
- nome torneio (Participa torneio)
- vitória (Participa ginásio)
- nome treinador (Participa ginásio)
- id ginásio (Participa ginásio)
- especialização (especialista)
- nome professor (Especialista)
- pokemon (Especialista)
- relacionamento (Relação ecológica)
- id pokemon 1 (relação ecológica)
- id pokemon 2 (relação ecológica)

- **CHECK:**

- gênero (Pokemon)
- tipo (Pokemon)
- vantagem (Pokemon)
- desvantagem(Pokemon)
- região (liga)
- liga responsável (Professor)
- liga responsável (Torneio)
- liga responsável (Ginásio)
- classificação (Participa Torneio)
- especialização (especialista)
- relacionamento (Relação ecológica)

- **CASCADE:**

- liga responsável (Professor)
- liga responsável (Torneio)
- liga responsável (Ginásio)
- treinador líder (Ginásio)
- nome torneio (Participa torneio)

- nome participante (Participa torneio)
 - id ginásio (Participa ginásio)
 - nome treinador (Participa ginásio)
 - nome professor (Especialista)
 - pokemon (Especialista)
 - id pokemon 1 (Relação ecológica)
 - id pokemon 2 (Relação ecológica)
- **UNIQUE:**
 - insígnia (Ginásio)
- **DEFAULT:**
 - número pokemon (Treinador)
 - nível (Pokemon)
 - classificação (Participa torneio)
 - relacionamento (Relação ecológica)
- **PRIMARY KEY:**
 - nome (Treinador)
 - id code (Pokémon)
 - nome (Liga)
 - nome (Professor)
 - nome (Torneio)
 - id ginásio (Ginásio)
 - num inscrição (Participa torneio)
 - num desafio (Participa ginásio)
 - id certificado (Especialista)
 - registro relacionamento (Relação ecológica)
- **SET NULL:**
 - orientador (Treinador)
 - treinador (Pokemon)

6. Consultas

Consulta 1 - “Quais pokémon um treinador possui?”

Aluno: Ariel Sadetsky

SQL:

```
SELECT nome
FROM pokemon
WHERE treinador LIKE <treinador>

/*<Treinador> representa qualquer elemento contido na
coluna treinador da tabela pokemon*/
```

Algebra relacional:

```
 $\pi_{\text{nome}}(\sigma(\text{treinador} = \text{<treinador>})(\text{pokemon}))$ 

/*<Treinador> representa qualquer elemento contido na
coluna treinador da tabela pokemon*/
```

	nome character varying (100) 🔒
1	Ditto
2	Pikachu
3	Mew

O resultado da consulta SQL pode ser visto acima, retornando uma tabela que possui uma coluna “nome” e 3 tuplas com os nomes dos pokémon pertencentes a determinado treinador.

Consulta 2 - “Quantas insígnias um treinador já possui?”

Aluno : Ariel Sadetsky

SQL:

```
SELECT numero_insignias
FROM treinador
WHERE nome LIKE <nome>

/*<nome> representa qualquer elemento contido na
coluna nome da tabela treinador*/
```

Algebra relacional:

```
 $\pi$  numero_insignias( $\sigma$ (nome = <nome>)(treinador))

/*<nome> representa qualquer elemento contido na
coluna nome da tabela treinador*/
```

	numero_insignias integer
1	4

O resultado da consulta SQL pode ser visto acima, retornando uma tabela que possui uma coluna “numero_insignias” e apenas 1 tupla (por conta da natureza da consulta que impossibilita o retorno de mais que uma tupla) com o número de insígnias que um determinado treinador possui.

Consulta 3 - “Se determinado treinador pode participar de um torneio”


Aluno: Danillo Santiago Oliveira da Conceição

SQL:

```
SELECT nome
FROM treinador
WHERE numero_insignias >= 4
```

Algebra relacional:

```
 $\pi_{\text{nome}}(\sigma(\text{numero\_insignias} \geq 4)(\text{treinador}))$ 
```

	nome [PK] character varying (100) 
1	Vanderlei
2	Gabriel
3	Erik

O resultado da consulta SQL pode ser visto acima, retornando uma tabela que possui uma coluna “nome” e 3 tuplas com os nomes dos treinadores aptos a participar de qualquer torneio.

Consulta 4 - “Qual o pokémon de menor nível de um determinado treinador?”

Aluno: Danillo Santiago Oliveira da Conceição

SQL:

```
SELECT id_code, nome, nivel
FROM pokemon
WHERE nivel IN (SELECT MIN(nivel)
                FROM pokemon
                WHERE treinador LIKE <treinador>
                GROUP BY treinador)

/*<Treinador> representa qualquer elemento contido na
coluna treinador da tabela pokemon*/
```

Algebra relacional:

```
T1 ← σ(treinador = <treinador>)(Pokemon)
T2 ← (P(min) F MIN(nível)(T1))
T3 ← Pokemon × T2
T4 ← πnome,id_code, nivel (σ(nivel = min)(T3))

/*<Treinador> representa qualquer elemento contido na
coluna treinador da tabela pokemon*/
```

	id_code [PK] integer	nome character varying (100)	nivel integer
1	222	Ditto	8
2	1000	Seal	8
3	245	Chinchar	8

O resultado da consulta SQL pode ser visto acima, retornando uma tabela que possui uma coluna “id_code”, outra coluna “nome”, uma coluna “nivel” e 3 tuplas com os nomes dos pokémon pertencentes a determinado treinador que possuem o menor nível. No caso, os 3 possuem o menor nível que é 8.

Consulta 5 - “Qual a média dos níveis dos pokémon de um determinado treinador?”

Aluno: Vanderlei Guilherme Andrade de Assis

SQL:


```
SELECT AVG(Nível)
FROM pokemon
WHERE treinador LIKE <treinador>

/*<Treinador> representa qualquer elemento contido na
coluna treinador da tabela pokemon*/
```

Algebra relacional:

```
T1 ←  $\sigma$ (treinador = <treinador>)(Pokemon)
T2 ← (média F avg(nível)(T1))

/*<Treinador> representa qualquer elemento contido na
coluna treinador da tabela pokemon*/
```

	avg numeric 
1	31.666666666666667

O resultado da consulta SQL pode ser visto acima, retornando uma tabela que possui uma coluna “avg” e apenas 1 tupla (por conta da natureza da consulta que impossibilita o retorno de mais que uma tupla) com a média dos níveis dos pokémon que um determinado treinador possui.

Consulta 6 - “Quantas vantagens um pokémon tem?”

Aluno: Vanderlei Guilherme Andrade de Assis

SQL:

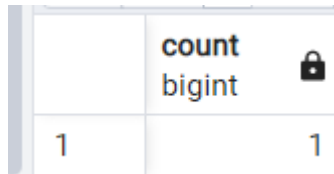
```
SELECT COUNT(Vantagem)
FROM pokemon
WHERE nome LIKE <nome>

/*<nome> representa qualquer elemento contido na
coluna nome da tabela pokemon*/
```

Algebra relacional:

```
T1 ← σ(nome = <nome>)(Pokémon)
F count(vantagem)(T1)

/*<nome> representa qualquer elemento contido na
coluna nome da tabela pokemon*/
```



	count
1	1

O resultado da consulta SQL pode ser visto acima, retornando uma tabela que possui uma coluna “count” e apenas 1 tupla (por conta da natureza da consulta que impossibilita o retorno de mais que uma tupla) com o número de vantagens que determinado pokémon possui. Vale ressaltar que, da forma como o projeto foi estruturado, um pokémon sempre terá 1 vantagem e uma fraqueza, tornando essa consulta desnecessária.

Consultas 7- “Quantos ginásios têm em determinada região”

Aluno: Vanderlei Guilherme Andrade de Assis

SQL:

```
SELECT COUNT(id_ginasio)
FROM ginasio
INNER JOIN Liga
ON liga_responsavel LIKE nome
WHERE regioao LIKE <regiao>
GROUP BY regioao

/*<regiao> representa qualquer elemento contido na
coluna regioao da tabela resultante da junção interna de liga e ginásio*?/
```

Algebra relacional:

```
t1 ← ((ginasio) ⋈ ginasio.liga_responsavel = liga.nome(liga))
t2 ← πregiao( σ(regiao = <região>)(t1))
F count(região)(t2)

/*<regiao> representa qualquer elemento contido na
coluna regioao da tabela resultante da junção interna de liga e ginásio*?/
```

	count bigint
1	1

O resultado da consulta SQL pode ser visto acima, retornando uma tabela que possui uma coluna “count” e apenas 1 tupla (por conta da natureza da consulta que impossibilita o retorno de mais que uma tupla) com o número de ginásios que estão em determinada região.

Consulta 8: “Qual relação ecológica uma espécie de pokémon tem com outros”

Aluno: Ariel Sadetsky

SQL:

```
SELECT P1.nome, P2.nome, relacionamento
FROM relação_ecologica
INNER JOIN pokemon P1
ON id_pokemon_1 = P1.id_code INNER JOIN pokemon P2 ON id_pokemon_2 = P2.id_code
WHERE id_pokemon_1 = 666

/*<P1.id_code> representa qualquer elemento contido na coluna id_code da tabela P1
(pokemon) e <P2.id_code> representa qualquer elemento contido na coluna id_code da tabela
P2 (pokemon)*/
```

Algebra relacional:

```
t1 ← nome_1 ρ nome( (Pokemon) ⋈ id_code = id_pokémon_1 (Relação_ecológica) )
t2 ← nome_2 ρ nome( (t1) ⋈ id_pokémon_2 = id_code (pokémon) )
t3 ← π nome_1, nome_2, relacionamento (σ id_pokemon_1 = <id_code> (t2))

/*<id_code> representa qualquer elemento contido na coluna id_code da tabela Pokémon*/
```

	nome character varying (100) 🔒	nome character varying (100) 🔒	relacionamento character varying (50) 🔒
1	Gengar	Pikachu	Harmônica
2	Gengar	Ditto	Amensalismo
3	Gengar	Abra	Competição

O resultado da consulta SQL pode ser visto acima, retornando uma tabela que possui duas colunas “nome” (uma para cada pokémon que está se relacionando) e 3 tuplas que mostram o mesmo pokémon “Gengar” se relacionando com outras 3 diferentes espécies e quais são essas relações.

Consulta 9 - “Quais e quantas vezes um treinador perdeu para um líder de ginásio ?”

Aluno: Danillo Santiago Oliveira da Conceição

SQL:

```
SELECT treinador_lider, COUNT(vitoria) as derrota
FROM Participa_ginasio NATURAL JOIN Ginasio
WHERE nome_treinador LIKE <nome_treinador> AND vitoria = false
GROUP BY treinador_lider, vitoria

/*<nome_treinador> representa qualquer elemento contido na
coluna nome_treinador da tabela resultante da junção interna de Participa_ginasio e
ginásio*/
```

Algebra Relacional:

```
T1 ← Participa_ginasio ⋈ ginasio
T2 ←  $\pi_{\text{treinador\_lider}}(\sigma(\text{nome\_treinador} = \text{<nome\_treinador>} \wedge \text{vitoria} = \text{false})(T1))$ 
T3 ← treinador_lider F count(treinador_lider)(T2)

/*<nome_treinador> representa qualquer elemento contido na
coluna nome_treinador da tabela resultante da junção interna de Participa_ginasio e
ginásio*/
```

	treinador_lider character varying (100) 🔒	derrotas bigint 🔒
1	Erik	2
2	Gabriel	1
3	Nicolas	1

O resultado da consulta SQL pode ser visto acima, retornando uma tabela que possui uma coluna “treinador_lider” e uma coluna “derrotas” além de 3 tuplas que relacionam o nome de líderes e quantidade de derrotas que determinado treinador sofreu para cada um deles.

7.Triggers - view

Foram formuladas duas triggers. A primeira foi feita para atualizar o número de insígnias de um treinador sempre que ele vence um ginásio, automatizando este processo. A segunda tem duas funções: primeiramente ela incrementa o número de pokémon que um treinador possui sempre que um é adicionado e, por fim, impede que um pokémon selvagem seja adicionado. Dessa forma, evita-se que a tabela pokémon seja poluída com dados desnecessários, uma vez que na criação da tabela, não se foi atribuída uma restrição NOT NULL na coluna treinador, já que se esperava certa relevância desses dados. Assim, corrigimos essa falha de projeto, de forma a não se precisar reestruturar novamente toda a modelagem do banco de dados, o que pode ser infactível em cenários em que o banco armazena gigas ou terabytes de informações. Também foi adicionada uma view ao projeto que apresenta uma tabela dos Pokémon e seus respectivos treinadores ordenados pelos nomes dos treinadores.

8. Considerações Finais

O grupo passou por algumas dificuldades durante o desenvolvimento do projeto. O escopo de nosso projeto foi o fator mais problemático, já que inicialmente nosso MER possuía 9 Entidades e mesmo após reformulações nos deparamos com um MR contendo 10 tabelas. Isso devido a uma certa riqueza de detalhes no minimundo no início do projeto. Com absoluta certeza, foi trabalhoso, difícil e às vezes inviável nos atermos a alguns detalhes de modelagem ou mesmo de implementação. Vale ressaltar que o resultado final atende à demanda inicial colocada em nosso projeto, descrevendo de fato uma ferramenta de categorização e catalogação dos monstros digitais (pokémon) e não “apenas” isso. O projeto veio a ser ainda maior que o planejamento inicial, sendo ele não só uma pokédex funcional (enciclopédia pokémon) como de boa parte de elementos que também estão contidos nesse universo, contendo um adicional de mecanismos que fizeram essa franquia de jogos prosperar, como treinadores, ginásios, ligas, professores, dentre outros. Mesmo com todos esses pontos listados, várias limitações ainda podem ser observadas, como por exemplo: A impossibilidade de se buscar a relação ecológica por espécies, apenas por Pokémon, a não possibilidade de se atribuir mais de 6 Pokémon a um treinador e o fato de não se poder listar todos os líderes que um treinador venceu junto com seu número total de vitórias sem causar redundância. Mesmo assim, estamos orgulhosos do resultado obtido e dos meios encontrados para que ele fosse alcançado..