

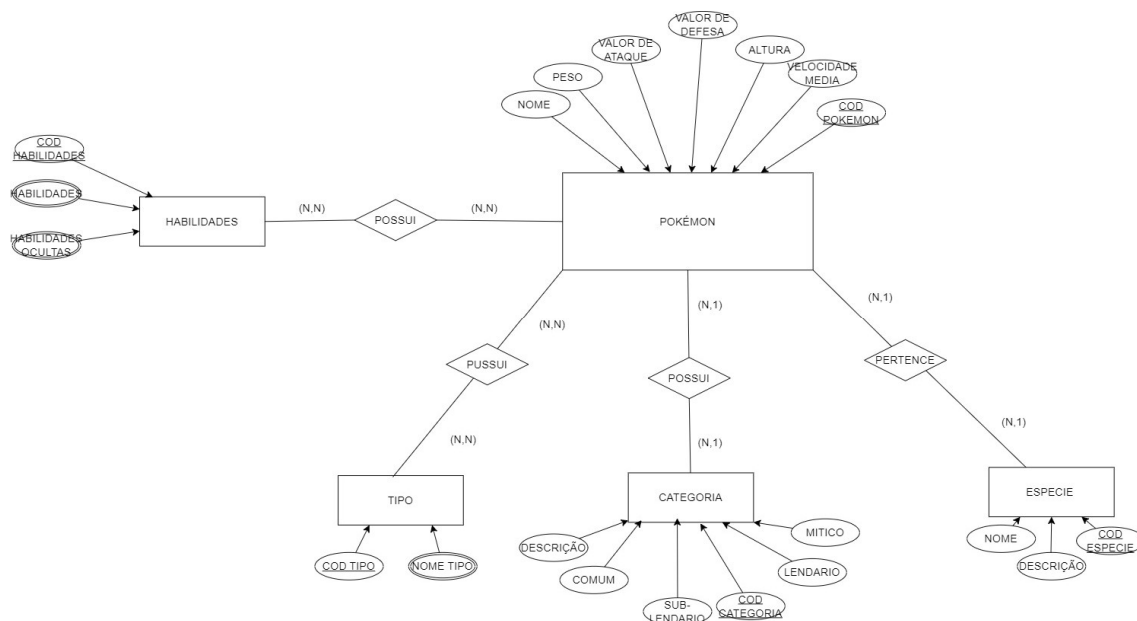
Relatório do Projeto - Pokédex

1. Introdução

Este relatório descreve o desenvolvimento do projeto da Pokédex, uma enciclopédia virtual com espécies de Pokémons. O objetivo do projeto foi modelar um banco de dados relacional que pudesse armazenar informações sobre os Pokémons, suas categorias, espécies, tipos e habilidades.

2. Diagrama de Entidades-Relacionamentos (D.E.R.)

O Diagrama de Entidades-Relacionamentos foi criado na etapa inicial do projeto e serviu como base para a criação do modelo relacional. O DER apresenta as entidades principais do sistema, seus relacionamentos e os atributos de cada entidade.



3. Modelo Relacional

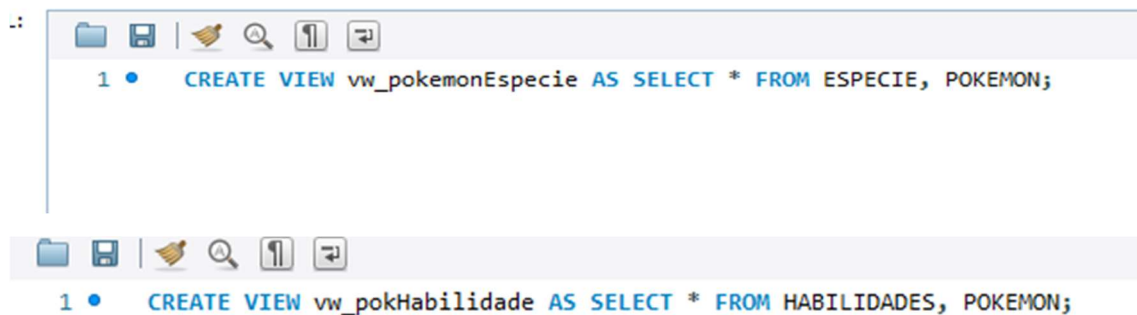
Com base no Diagrama de Entidades-Relacionamentos, foi elaborado o modelo relacional do banco de dados. O modelo relacional define as tabelas, seus atributos e as chaves primárias e estrangeiras.

4. Comandos SQL-DDL

Os comandos SQL-DDL foram utilizados para criar as tabelas no banco de dados específico do grupo. As tabelas foram criadas visualmente e os comandos foram gerados pelo ambiente de desenvolvimento. As chaves primárias e estrangeiras foram definidas nos comandos de criação das tabelas.

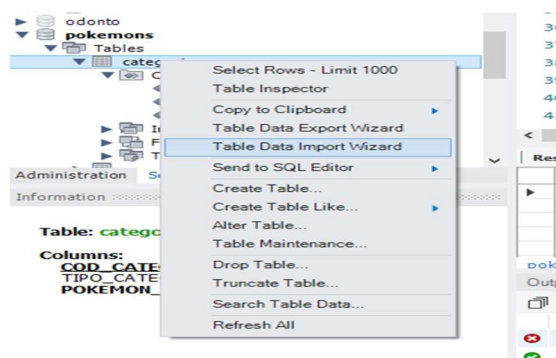
5. Comandos para Criação de Visões

Foram criados dois comandos para a criação de visões no banco de dados. As visões foram utilizadas para fornecer diferentes perspectivas e facilitar consultas específicas aos dados armazenados.



6. Inserção de Pokémons

Para popular o banco de dados com informações sobre os Pokémons, foram utilizados os dados disponíveis no conjunto de dados fornecido em: [\[INSERIR LINK PARA O CONJUNTO DE DADOS\]](#). Pelo menos 50 Pokémons foram inseridos no banco de dados, com base nessas informações.



7. Consultas SQL

Foram desenvolvidas 10 consultas SQL criativas para extrair informações relevantes do banco de dados. Cada consulta segue uma especificação diferente, explorando junções, filtros, funções agregadas e subconsultas. A seguir, são apresentadas as consultas realizadas:

-- A. Junções de 2 ou mais tabelas, com ORDER BY:

```
SELECT *  
  
FROM pokemon  
  
JOIN categoria ON pokemon.COD_POK = categoria.COD_CATEGORIA  
  
ORDER BY pokemon.COD_POK, categoria.COD_CATEGORIA;
```

-- B. Junções de 2 ou mais tabelas, com ORDER BY e filtros na cláusula WHERE:

```
SELECT *  
  
FROM pokemon  
  
JOIN especie ON pokemon.COD_POK = especie.COD_ESPECIE  
  
WHERE especie.NOME_ESPECIE = 'Weed PokÃfÃ©mon'  
  
ORDER BY pokemon.COD_POK, especie.COD_ESPECIE;
```

-- C. Junções de 3 ou mais tabelas, com ORDER BY e filtros na cláusula WHERE:

```
select * from pokemon  
  
join habilidades on pokemon.COD_POK = habilidades.COD_HAB  
  
join tipo on habilidades.COD_HAB = tipo.COD_TIPO  
  
where tipo.NOME_TIPO = 'Grass'  
  
order by pokemon.COD_POK, habilidades.COD_HAB, tipo.COD_TIPO;
```

-- D. Junção de 3 ou mais tabelas, usando os operadores LIKE e BETWEEN:

```
SELECT *  
  
FROM pokemon  
  
JOIN categoria ON pokemon.COD_POK = categoria.COD_CATEGORIA  
  
JOIN habilidades ON categoria.COD_CATEGORIA = habilidades.COD_HAB  
  
WHERE pokemon.NOME LIKE '%Bulbasaur%' AND pokemon.VALOR_ATAQUE BETWEEN 20 AND  
50
```

ORDER BY pokemon.nome, categoria.TIPO_CATEGORIA, habilidades.HABILIDADES;

select * FROM categoria;

-- E. Junção de 2 ou mais tabelas, usando os operadores IN e IS NULL/IS NOT NULL:

SELECT *

FROM pokemon

JOIN categoria ON pokemon.COD_POK = categoria.COD_CATEGORIA IN ('Grass', 'fire') AND
categoria.TIPO_CATEGORIA IS NOT NULL

ORDER BY pokemon.COD_POK, categoria.COD_CATEGORIA;

-- F. Junção de 2 ou mais tabelas com GROUP BY, sem HAVING, usando uma função agregada
qualquer (MIN, MAX, AVG, SUM, COUNT). Use ORDER BY:

SELECT *

FROM pokemon

JOIN categoria ON pokemon.COD_POK = categoria.COD_CATEGORIA

GROUP BY pokemon.COD_POK

ORDER BY pokemon.COD_POK

-- G. Junção de 2 ou mais tabelas com GROUP BY e HAVING, usando uma função agregada
qualquer (MIN, MAX, AVG, SUM, COUNT):

SELECT categoria.TIPO_CATEGORIA, COUNT(*) AS nome

FROM pokemon

JOIN categoria ON pokemon.COD_POK = categoria.COD_CATEGORIA

JOIN tipo ON pokemon.COD_POK = COD_TIPO

GROUP BY categoria.TIPO_CATEGORIA

HAVING COUNT(*) > 5;

-- H. Subselect sem correlação:

SELECT nome

FROM pokemon

WHERE categoria.COD_CATEGORIA IN (SELECT COD_CATEGORIA FROM categoria WHERE
TIPO_CATEGORIA = 'Fire');

-- I. Subselect com correlação:

SELECT nome

FROM pokemon p

WHERE EXISTS (SELECT 1 FROM habilidades h WHERE h.COD_HAB = p.COD_POK AND
h.NUMEROHABILIDADES = '2');

-- J. Subselect com EXISTS:

SELECT nome

FROM pokemon

WHERE EXISTS (SELECT 1 FROM categoria WHERE categoria.COD_CATEGORIA =
pokemon.COD_POK);

8. Conclusão

O projeto da Pokédex foi concluído com sucesso, atendendo aos requisitos de modelagem do banco de dados e permitindo consultas eficientes sobre as informações dos Pokémons. O banco de dados relacional proporcionou uma estrutura organizada para armazenar e consultar os dados relacionados a cada espécie de Pokémon, suas categorias, tipos e habilidades.