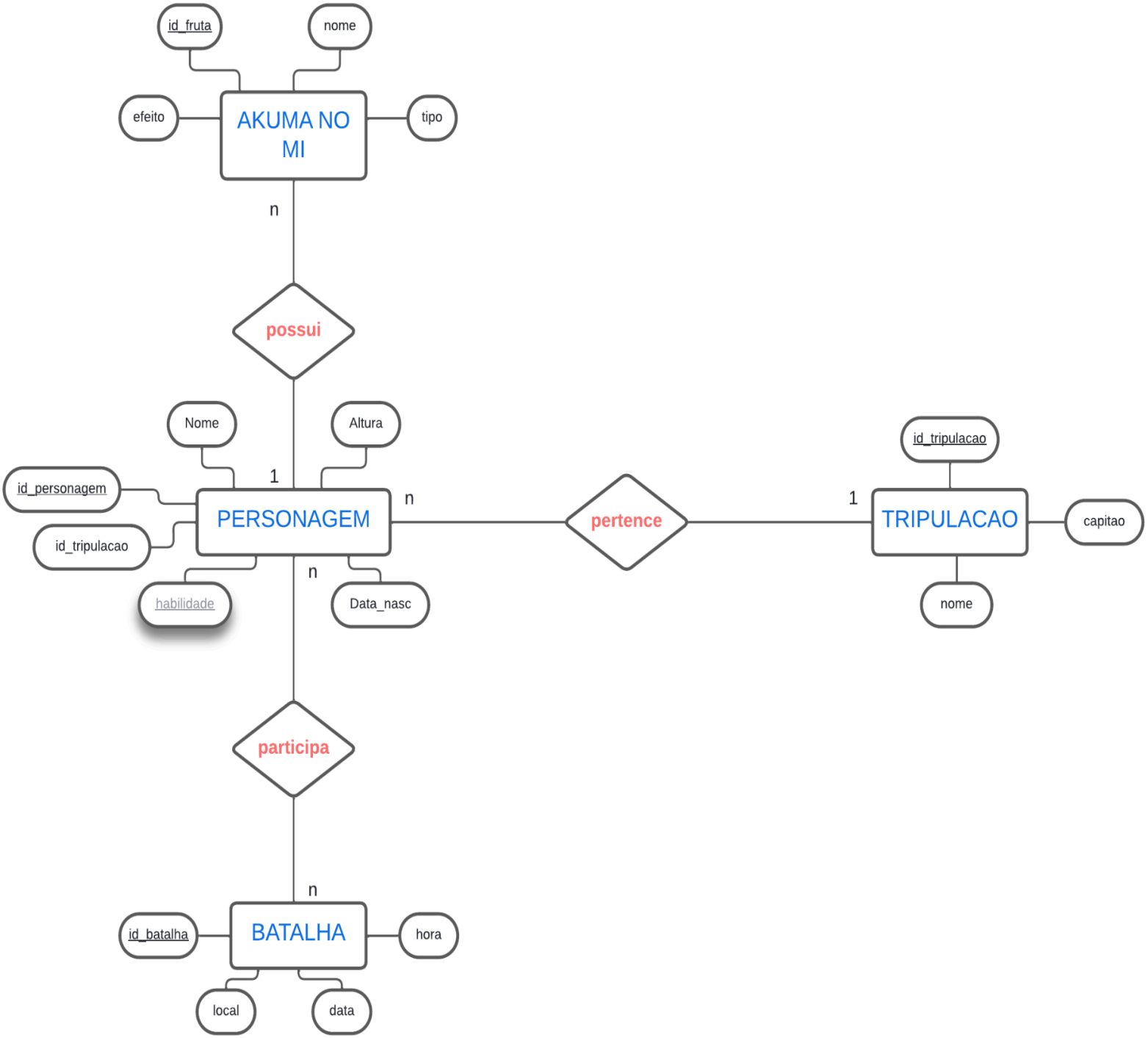
Lista 4

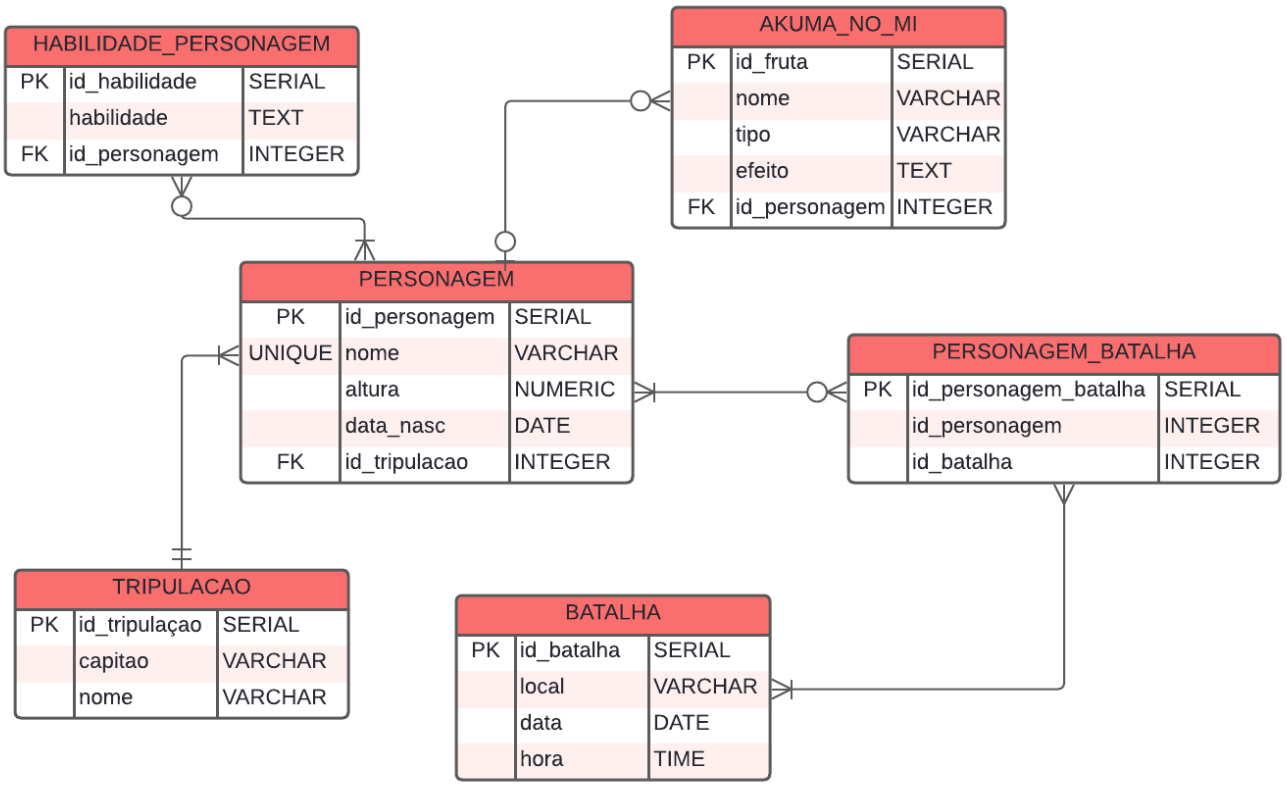
Ana Clara e Lívia Braz

Modelo Entidade-Relacionamento (MER)



Campo Multivalorado: Habilidade do Personagem

Modelo Relacional (MR)



DDL (Data Definition Language)

-- Criando as tabelas: 2 com chaves estrangeiras, 1 coluna com NOT NULL em cada tabela, 1 coluna com DEFAULT, SERIAL, DATE, TIME, NUMERIC, INTEGER, VARCHAR, outro tipo

```
CREATE TABLE tripulacao (
    id SERIAL PRIMARY KEY,
    nome VARCHAR(100) NOT NULL,
    capitao VARCHAR(100) DEFAULT 'Desconhecido'
);

CREATE TABLE akuma_no_mi (
    id SERIAL PRIMARY KEY,
    nome VARCHAR(100) NOT NULL,
    tipo VARCHAR(50),
    efeito TEXT,
    id_personagem INTEGER,
    CONSTRAINT fk_personagem FOREIGN KEY (id_personagem) REFERENCES personagem(id),
    UNIQUE (nome)
);
CREATE TABLE batalha (
    id SERIAL PRIMARY KEY,
    local VARCHAR(100) NOT NULL,
    data DATE NOT NULL,
    hora TIME
);
```

```
CREATE TABLE personagem (
    id SERIAL PRIMARY KEY,
    nome VARCHAR(100) NOT NULL UNIQUE,
    data_nasc DATE,
    altura NUMERIC,
    id_tripulacao INTEGER,
    CONSTRAINT fk_tripulacao FOREIGN KEY (id_tripulacao) REFERENCES tripulacao(id)
);

CREATE TABLE personagem_batalha (
    id_personagem_batalha SERIAL NOT NULL PRIMARY KEY,
    id_personagem INTEGER NOT NULL,
    id_batalha INTEGER NOT NULL,
    CONSTRAINT fk_personagem FOREIGN KEY (id_personagem) REFERENCES personagem(id),
    CONSTRAINT fk_batalha FOREIGN KEY (id_batalha) REFERENCES batalha(id)
);

CREATE TABLE habilidade_personagem (
    id_habilidade INTEGER NOT NULL PRIMARY KEY,
    habilidade TEXT,
    id_personagem INTEGER NOT NULL,
    CONSTRAINT fk_personagem FOREIGN KEY (id_personagem) REFERENCES personagem(id)
);

-- criação de coluna em uma tabela já existente
ALTER TABLE personagem ADD COLUMN recompensa NUMERIC;

-- exclusão de coluna em uma tabela já existente.
ALTER TABLE personagem DROP COLUMN altura;

-- alteração do nome de uma coluna
ALTER TABLE personagem_batalha RENAME COLUMN id_personagem_batalha TO id_pers_bat;

-- coluna como UNIQUE em uma tabela já existente
ALTER TABLE akuma_no_mi ADD CONSTRAINT unique_nome UNIQUE (nome);

-- exclusão de restrição UNIQUE
ALTER TABLE akuma_no_mi DROP CONSTRAINT unique_nome;

-- modificação de valor DEFAULT
ALTER TABLE tripulacao ALTER COLUMN capitao SET DEFAULT ' ';

-- modificação do tipo de uma coluna
ALTER TABLE batalha ALTER COLUMN local SET DATA TYPE TEXT;

-- alteração do nome de uma tabela
ALTER TABLE personagem_batalha  RENAME TO pers_batalha;
```

```
-- ALGUNS INSERTS PARA POPULAR A TABELA
INSERT INTO tripulacao (nome, capitao)
VALUES ('Piratas do Chapéu de Palha', 'Luffy'), ('Piratas do Coração',''), ('Marinha'," ), ('Piratas do Barba Branca', 'Barba Branca'), ('Piratas do Kid', 'Kid'), ('Piratas do Law', 'Law'), ('Piratas do Buggy', 'Buggy'), ('Piratas do Shanks', 'Shanky'), ('Piratas do Barba Negra', 'Barba Negra'), ('Revolucionários'," );

INSERT INTO akuma_no_mi (nome, tipo, efeito, id_personagem)
VALUES
        ('Gomu Gomu no Mi', 'Paramecia', 'Transforma o corpo em borracha', 1),
        ('Ope Ope no Mi', 'Paramecia', 'Permite criar um "quarto" e manipular tudo dentro dele', 11),
        ('Mera Mera no Mi', 'Logia', 'Permite criar e controlar fogo', 12),
        ('Hie Hie no Mi', 'Logia', 'Permite criar e controlar gelo', 21),
        ('Magu Magu no Mi', 'Logia', 'Permite criar e controlar magma', 22),
        ('Hana Hana no Mi', 'Paramecia', 'Permite fazer partes do corpo brotar de qualquer superfície', 5);

INSERT INTO batalha (local, data, hora) VALUES
        ('Marineford', '2022-01-01', '12:00'),
        ('Enies Lobby', '2022-02-01', '14:00'),
        ('Sabaody', '2022-03-01', '16:00'),
        ('Punk Hazard', '2022-04-01', '10:00'),
        ('Dressrosa', '2022-05-01', '18:00'),
        ('Whole Cake Island', '2022-06-01', '09:00'),
        ('Wano', '2022-07-01', '11:00'),
        ('Alabasta', '2022-08-01', '13:00'),
        ('Skypiea', '2022-09-01', '15:00'),
        ('Fishman Island', '2022-10-01', '17:00');

INSERT INTO personagem (nome, data_nasc, id_tripulacao, recompensa)
VALUES
        ('Monkey D. Luffy', '1997-05-05', 1, 1500000000),
        ('Roronoa Zoro', '1998-11-11', 1, 320000000),
        ('Nami', '1999-07-03', 1, 66000000),
        ('Usopp', '1999-04-01', 1, 200000000),
        ('Sanji', '1999-03-02', 1, 330000000),
        ('Tony Tony Chopper', '2000-12-24', 1, 100),
        ('Nico Robin', '1998-02-06', 1, 130000000),
        ('Franky', '1998-03-09', 1, 94000000),
        ('Brook', '1997-04-03', 1, 83000000),
        ('Jinbe', '1996-03-02', 1, 438000000),
        ('Trafalgar D. Water Law', '1996-10-06', 2, 500000000),
        ('Portgas D. Ace', '1997-01-01', 4, 550000000),
        ('Sakazuki', '1990-08-16', 3, 0),
        ('Borsalino', '1991-11-23', 3, 0),
        ('Kuzan', '1992-09-21', 3, 0),
        ('Smoker', '1993-03-14', 3, 0),
        ('Eustass Kid', '1997-01-10', 5, 470000000),
        ('Dracule Mihawk', '1985-03-09', 6, 0),
        ('Boa Hancock', '1995-09-02', 7, 800000000),
        ('Shanks', '1986-03-09', 8, 4000000000),
        ('Marshall D. Teach', '1980-08-03', 9, 2247600000),
        ('Sabo', '1998-03-20', 10, 602000000);

INSERT INTO pers_batalha (id_personagem, id_batalha)
VALUES
        (1, 1), (2, 1), (3, 1), (4, 1), (5, 1), (1, 2), (6, 2), (7, 2), (8, 2), (9, 2),
        (10, 2), (11, 3), (12, 3), (13, 3), (14, 3), (15, 3), (16, 3), (17, 3), (18, 4),
        (19, 4), (20, 4), (21, 4), (22, 5), (1, 5), (2, 5), (3, 5), (4, 5);

INSERT INTO habilidade_personagem (id_habilidade, habilidade, id_personagem)
VALUES
        (1, 'Gomu Gomu no Pistol', 1), (2, 'Santoryu', 2), (3, 'Clima-Tact', 3), (4, 'Sniper King', 4), (5, 'Diable Jambe', 5),
        (6, 'Heavy Point', 6), (7, 'Cien Fleurs', 7), (8, 'Franky Shogun', 8), (9, 'Soul Solid', 9), (10, 'Fishman Karate', 10),
        (11, 'Room', 11), (12, 'Hiken', 12), (13, 'Magma Fist', 13), (14, 'Yasakani no Magatama', 14), (15, 'Ice Age', 15),
        (16, 'White Out', 16), (17, 'Punk Rotten', 17), (18, 'Black Blade', 18), (19, 'Mero Mero Mellow', 19), (20, 'Haki', 20),
        (21, 'Blackbeard Power', 21), (22, 'Dragon Claw', 22);
```

DML (Data Manipulation Language)

```
-- INSERT escrevendo as colunas
INSERT INTO tripulacao (nome, capitao) VALUES ('Piratas do Big Mom', 'Big Mom');

-- INSERT não escrevendo as colunas
INSERT INTO tripulacao VALUES (DEFAULT, 'Piratas');

-- INSERT com RETURNING
INSERT INTO personagem (nome, data_nasc, id_tripulacao, recompensa)
VALUES ('Cavendish', '1997-07-31', 6, 330000000) RETURNING id;

-- UPDATE
UPDATE personagem SET recompensa = 200000000 WHERE nome = 'Usopp';

-- UPDATE com IN
UPDATE personagem SET recompensa = 1000000000 WHERE id IN (11, 12, 21);

-- UPDATE com NOT IN
UPDATE personagem SET recompensa = 50000000 WHERE id NOT IN (1, 2, 3, 4, 5);

-- DELETE
DELETE FROM personagem WHERE nome = 'Cavendish';

-- DELETE com WHERE
DELETE FROM personagem WHERE id = 24;

-- SELECT com ORDER BY com ordenação ASC
SELECT nome FROM personagem ORDER BY nome ASC;

-- SELECT com ORDER BY com ordenação DESC
SELECT nome FROM personagem ORDER BY nome DESC;

-- SELECT com GROUP BY
SELECT id_tripulacao, COUNT(*) AS numero_personagens FROM personagem GROUP BY
id_tripulacao;

-- SELECT com GROUP BY com uma função de agregação
SELECT id_tripulacao, SUM(recompensa) AS total_recompensa FROM personagem GROUP BY
id_tripulacao;

-- SELECT com usando a função de agregação MAX
SELECT MAX(recompensa) AS recompensa_maxima FROM personagem;

-- SELECT com usando a função de agregação MIN
SELECT MIN(recompensa) AS recompensa_minima FROM personagem;

-- SELECT com usando a função de agregação COUNT
SELECT COUNT(*) AS total_personagens FROM personagem;

-- SELECT com usando a função de agregação AVG
SELECT AVG(recompensa) AS recompensa_media FROM personagem;

-- SELECT com usando a função de agregação SUM
SELECT SUM(recompensa) AS recompensa_total FROM personagem;
```

```sql
-- SELECT com DISTINCT
SELECT DISTINCT id_tripulacao FROM personagem;

-- SELECT com AND
SELECT nome FROM personagem WHERE id_tripulacao = 1 AND recompensa > 100000000;

-- SELECT com OR
SELECT nome FROM personagem WHERE id_tripulacao = 1 OR recompensa > 1000000000;

-- SELECT com NOT
SELECT nome FROM personagem WHERE NOT id_tripulacao = 3;

-- SELECT com BETWEEN
SELECT nome FROM personagem WHERE recompensa BETWEEN 100000000 AND 500000000;

-- SELECT com INNER JOIN
SELECT t.nome AS nome_tripulacao, string_agg(p.nome, ', ') AS nome FROM personagem p INNER JOIN tripulacao t ON p.id_tripulacao = t.id GROUP BY 1;

-- SELECT com LEFT JOIN
SELECT t.nome AS nome_tripulacao, string_agg(p.nome, ', ') AS nome FROM personagem p LEFT JOIN tripulacao t ON p.id_tripulacao = t.id GROUP BY 1;

-- SELECT com RIGHT JOIN
SELECT t.nome AS nome_tripulacao, string_agg(p.nome, ', ') AS nome FROM personagem p RIGHT JOIN tripulacao t ON p.id_tripulacao = t.id GROUP BY 1;

-- SELECT com FULL OUTER JOIN
SELECT t.nome AS nome_tripulacao, string_agg(p.nome, ', ') AS nome FROM personagem p FULL OUTER JOIN tripulacao t ON p.id_tripulacao = t.id GROUP BY 1;

-- SELECT usando o LIMIT
SELECT nome FROM personagem LIMIT 5;

-- SELECT usando o OFFSET
SELECT nome FROM personagem OFFSET 5;

-- SELECT substituindo os valores nulos no retorno
SELECT nome, COALESCE(recompensa, 0) AS recompensa FROM personagem;

-- SELECT buscando dados diferentes de algo
SELECT nome FROM personagem WHERE id_tripulacao <> 3;

-- SELECT com LIKE
SELECT nome FROM personagem WHERE nome LIKE 'Monkey%';

-- SELECT com NOT LIKE
SELECT nome FROM personagem WHERE nome NOT LIKE 'Monkey%';

-- SELECT com CASE
SELECT nome,
   CASE
      WHEN recompensa > 1000000000 THEN 'Alta'
      WHEN recompensa BETWEEN 100000000 AND 1000000000 THEN 'Média'
      ELSE 'Baixa'
   END AS categoria_recompensa
FROM personagem;
```

```
-- SELECT com round
SELECT nome, ROUND(recompensa, -6) AS recompensa_arredondada FROM personagem;

-- SELECT com trunc
SELECT nome, TRUNC(recompensa, -6) AS recompensa_truncada FROM personagem;

-- SELECT com UPPER
SELECT UPPER(nome) AS nome_maiusculo FROM personagem;

-- SELECT com LOWER
SELECT LOWER(nome) AS nome_minusculo FROM personagem;

-- SELECT com substring
SELECT nome, SUBSTRING(nome FROM 1 FOR 3) AS iniciais FROM personagem;

-- SELECT com conversão de dados
SELECT nome, CAST(recompensa AS VARCHAR) AS recompensa_texto FROM personagem;

-- SELECT com concatenação de string
SELECT nome, 'Recompensa: ' || recompensa AS detalhes_recompensa
FROM personagem;

-- SELECT com a função “age”
SELECT nome, AGE(data_nasc) AS idade
FROM personagem;
```