

Trabalho - SQL | Fear & Hunger

Grupo: Fear & Hunger

Integrantes:

- Alexandre de Santana Beck - 211061350
- Gabriela de Oliveira Lemos - 170010872
- Matheus Fonseca Sousa - 170111059
- Pedro Lima - 170020525

Link documentação: https://sbd1.github.io/2023.2_Fear_and_Hunger/#!/

Link Repositorio: https://github.com/SBD1/2023.2_Fear_and_Hunger/

Link Apresentação: <https://www.youtube.com/watch?v=rAxV4Q2LYVc>

Atualizações:

- DER: Ajustes nos atributos
 - https://sbd1.github.io/2023.2_Fear_and_Hunger/#!/modulo_01/der
- Modelo Entidade Relacionamento: Atualização da via atualização com base no DER
 - https://sbd1.github.io/2023.2_Fear_and_Hunger/#!/modulo_01/modeloEntidadeRelacionamento
- Dicionário de Dados: Atualizando tabelas conforme a nova versão do MER
 - https://sbd1.github.io/2023.2_Fear_and_Hunger/#!/modulo_01/dicionarioDeDados

Criação das Tabelas

Tabela Personagem

```
CREATE TABLE Personagem (  
  idPersonagem INT PRIMARY KEY AUTO_INCREMENT,  
  alma INT DEFAULT 0,  
  local INT UNIQUE,  
  tipoP INT NOT NULL  
);
```

Tabela Loot

```
CREATE TABLE Loot (  
  FOREIGN KEY (local) REFERENCES Local (idLocal),  
  FOREIGN KEY (item) REFERENCES Item (idItem)  
);
```

Tabela Coletadas

```
CREATE TABLE Coletadas (  
  FOREIGN KEY (personagem) REFERENCES PersonagemJogavel (idPersonagem),  
  FOREIGN KEY (alma) REFERENCES Alma (nome),  
);
```

Tabela Ataque

```
CREATE TABLE Ataque (  
  idAtaque INT PRIMARY KEY AUTO_INCREMENT,  
  tipoAtaque INT NOT NULL,  
  descrição VARCHAR(255) DEFAULT '',  
  FOREIGN KEY(idHabilidade) REFERENCES Habilidade (idHabilidade),  
  danoFisico INT DEFAULT 0,  
  danoMagico INT DEFAULT 0  
);
```

Tabela Item

```
CREATE TABLE Item (  
  idItem INT PRIMARY KEY AUTO_INCREMENT,  
  tipo VARCHAR(255) NOT NULL,  
  FOREIGN KEY(idInventario) REFERENCES Inventario (idInventario),  
  FOREIGN KEY(lojista) REFERENCES Lojista (idLojista)  
);
```

Tabela Armadura

```
CREATE TABLE Armadura (  
  FOREIGN KEY (idItem) REFERENCES Item (idItem),  
  defesa INT DEFAULT 0,  
  defesaMagica INT DEFAULT 0,  
  agilidade INT DEFAULT 0,  
  descrição VARCHAR(255) DEFAULT '',  
  valor INT DEFAULT 0,  
  nome VARCHAR(255) NOT NULL  
);
```

Tabela Local

```
CREATE TABLE Local (  
  idLocal PRIMARY KEY AUTO_INCREMENT  
);
```

Tabela Estoque

```
CREATE TABLE Estoque (  
  FOREIGN KEY (idLojista) REFERENCES Lojista (idLojista),  
  FOREIGN KEY (idItem) REFERENCES Item (idItem)  
);
```

Tabela Arma

```
CREATE TABLE Arma (
  FOREIGN KEY (idItem) REFERENCES Item (idItem),
  ataqueFisico INT DEFAULT 0,
  ataqueMagico INT DEFAULT 0,
  descrição VARCHAR(255) DEFAULT '',
  valor INT DEFAULT 0,
  nome VARCHAR(255) NOT NULL
);
```

Tabela Acessorio

```
CREATE TABLE Acessorio (
  FOREIGN KEY (idItem) REFERENCES Item (idItem),
  defesa INT DEFAULT 0,
  defesaMagica INT DEFAULT 0,
  agilidade INT DEFAULT 0,
  ataqueFisico INT DEFAULT 0,
  ataqueMagico INT DEFAULT 0,
  descrição VARCHAR(255) DEFAULT '',
  valor INT DEFAULT 0,
  nome VARCHAR(255) NOT NULL
);
```

Tabela Lojista

```
CREATE TABLE Lojista (
  idLojista INT PRIMARY KEY AUTO_INCREMENT,
  FOREIGN KEY (idInventario) REFERENCES Inventario (idInventario)
  agilidade INT DEFAULT 0,
  defesaMagica INT DEFAULT 0,
  defesa INT DEFAULT 0,
  ataque INT DEFAULT 0,
  nome VARCHAR(255) NOT NULL
);
```

Tabela Personagem Não Jogável

```
CREATE TABLE PNJ (
  FOREIGN KEY (idPersonagem) REFERENCES Personagem (idPersonagem),
  tipoPnj NOT NULL
);
```

Tabela Inimigo

```
CREATE TABLE Inimigo (
  idPersonagem INT PRIMARY KEY AUTO_INCREMENT,
  defesa INT DEFAULT 0,
  agilidade INT DEFAULT 0,
  ataque INT DEFAULT 0,
  nome VARCHAR(255) NOT NULL,
  defesaMagica INT DEFAULT 0
);
```

Tabela Legivel

```
CREATE TABLE Legivel (
  idItem INT PRIMARY KEY AUTO_INCREMENT,
  conteúdo VARCHAR(255) DEFAULT '',
  valor INT DEFAULT 0,
  descrição VARCHAR(255) DEFAULT '',
  nome VARCHAR(255) NOT NULL
);
```

Tabela Consumivel

```
CREATE TABLE Consumivel (
  idItem INT PRIMARY KEY AUTO_INCREMENT,
  adHp INT NOT NULL,
  adMente INT NOT NULL,
  descrição VARCHAR(255) DEFAULT '',
  valor INT DEFAULT 0,
  nome VARCHAR(255) NOT NULL
);
```

Tabela Inventario

```
CREATE TABLE Inventario (
  idInventario INT PRIMARY KEY AUTO_INCREMENT,
  FOREIGN KEY (idPersonagem) REFERENCES Personagem (idPersonagem)
);
```

Tabela Alma

```
CREATE TABLE Alma (
  nome VARCHAR(255) PRIMARY KEY,
  FOREIGN KEY (personagem) REFERENCES Personagem (idPersonagem)
);
```

Tabela Personagem Jogável

```
CREATE TABLE PersonagemJogavel (
  corpo INT NOT NULL,
  mente INT NOT NULL,
  defesa INT NOT NULL,
  agilidade INT NOT NULL,
  ataque INT NOT NULL,
  nome VARCHAR(255) NOT NULL,
  defesaMagica INT NOT NULL,
  FOREIGN KEY (idPersonagem) REFERENCES Personagem (idPersonagem),
  FOREIGN KEY (acessorio1) REFERENCES Item (idItem),
  FOREIGN KEY (acessorio2) REFERENCES Item (idItem),
  FOREIGN KEY (Arma) REFERENCES Item (idItem),
  FOREIGN KEY (Armadura) REFERENCES Item (idItem)
);
```

Tabela Habilidade

```
CREATE TABLE Habilidade (  
  idHabilidade INT PRIMARY KEY AUTO_INCREMENT,  
  custo INT NOT NULL,  
  FOREIGN KEY (alma) REFERENCES Alma (Nome)  
);
```

Tabela Genital

```
CREATE TABLE Genital (  
  idParte INT PRIMARY KEY,  
  FOREIGN KEY (Personagem) REFERENCES Personagem (idPersonagem),  
  tipoParte INT UNIQUE,  
  hpMaximo INT NOT NULL,  
  hitChance INT NOT NULL,  
  hpAtual INT NOT NULL  
);
```

Tabela Cabeça

```
CREATE TABLE Cabeça (  
  idParte INT PRIMARY KEY,  
  FOREIGN KEY (Personagem) REFERENCES Personagem (idPersonagem),  
  tipoParte INT UNIQUE,  
  hpMaximo INT NOT NULL,  
  hitChance INT NOT NULL,  
  hpAtual INT NOT NULL  
);
```

Tabela Olho

```
CREATE TABLE Olho (  
  idParte INT PRIMARY KEY,  
  FOREIGN KEY (Personagem) REFERENCES Personagem (idPersonagem),  
  tipoParte INT UNIQUE,  
  hpMaximo INT NOT NULL,  
  hitChance INT NOT NULL,  
  hpAtual INT NOT NULL  
);
```

Tabela Braço

```
CREATE TABLE Braco (  
  idParte INT PRIMARY KEY,  
  FOREIGN KEY (Personagem) REFERENCES Personagem (idPersonagem),  
  tipoParte INT UNIQUE,  
  hpMaximo INT NOT NULL,  
  hitChance INT NOT NULL,  
  hpAtual INT NOT NULL  
);
```

Tabela Torco

```
CREATE TABLE Torco (
  idParte INT PRIMARY KEY,
  FOREIGN KEY (Personagem) REFERENCES Personagem (idPersonagem),
  tipoParte INT UNIQUE,
  hpMaximo INT NOT NULL,
  hitChance INT NOT NULL,
  hpAtual INT NOT NULL
);
```

Tabela *Perna*

```
CREATE TABLE Perna (
  idParte INT PRIMARY KEY,
  FOREIGN KEY (Personagem) REFERENCES Personagem (idPersonagem),
  tipoParte INT UNIQUE,
  hpMaximo INT NOT NULL,
  hitChance INT NOT NULL,
  hpAtual INT NOT NULL
);
```

Inserção de Dados

Entidade Alma:

```
INSERT INTO Alma(nome, personagem) VALUES ('Liberdade', 'NULL' );

INSERT INTO Alma(nome, personagem) VALUES('Sombria', 'NULL');

INSERT INTO Alma(nome, personagem) VALUES('Feral', 'NULL');

INSERT INTO Alma(nome, personagem) VALUES('Caótica', 'NULL');
```

Entidade Personagem:

```
INSERT INTO Personagem(idPersonagem, alma, local, tipoP) VALUES('NI01', 'Feral', 'NULL', 'Inimigo');
```

Entidade Personagem Não Jogável:

```
INSERT INTO PersonagemNaoJogavel(idPnj, tipoPnj) VALUES ('NL01', 'L');

INSERT INTO PersonagemNaoJogavel(idPnj, tipoPnj) VALUES ('NI01', 'I');

INSERT INTO PersonagemNaoJogavel(idPnj, tipoPnj) VALUES ('NI02', 'I');

INSERT INTO PersonagemNaoJogavel(idPnj, tipoPnj) VALUES ('NI03', 'I');

INSERT INTO PersonagemNaoJogavel(idPnj, tipoPnj) VALUES ('NI04', 'I');

INSERT INTO PersonagemNaoJogavel(idPnj, tipoPnj) VALUES ('NI05', 'I');

INSERT INTO PersonagemNaoJogavel(idPnj, tipoPnj) VALUES ('NI06', 'I');
```

Entidade Lojista:

```
INSERT INTO Lojista(idLojista, agilidade, defesaMagica, defesa, ataque, nome) VALUES('NL01', 11, 40, 15, 40, 'Sinistro');
```

Entidade Inimigo:

```
INSERT INTO Inimigo(idInimigo, defesa, ataque, nome, tipoPnj, agilidade) VALUES ('NI01', 15, 25, 'Ghoul', 'Inimigo', 11);

INSERT INTO Inimigo(idInimigo, defesa, ataque, nome, tipoPnj, agilidade) VALUES ('NI02', 10, 10, 'Chamuscad o', 'Inimigo', 11);

INSERT INTO Inimigo(idInimigo, defesa, ataque, nome, tipoPnj, agilidade) VALUES('NI03', 20, 35, 'Padre Decréto', 'Inimigo', 11);

INSERT INTO Inimigo(idInimigo, defesa, ataque, nome, tipoPnj, agilidade) VALUES('NI04', 25, 40, 'Zelador ', 'Inimigo', 14);

INSERT INTO Inimigo(idInimigo, defesa, ataque, nome, tipoPnj, agilidade) VALUES('NI05', 20, 35, 'Cão ', 'Inimigo', 11);

INSERT INTO Inimigo(idInimigo, defesa, ataque, nome, tipoPnj, agilidade) VALUES('NI06', 40, 45, 'Palhaço ', 'Inimigo', 15);
```

Entidade Armadura:

```
INSERT INTO Armadura (idItem, defesa, defesaMagica, agilidade, descricao, valor, nome)
VALUES (1, 1, 0, 1, 'Armadura de couro feita de peles de animais selvagens.', 50, 'Couro');

INSERT INTO Armadura (idItem, defesa, defesaMagica, agilidade, descricao, valor, nome)
VALUES (2, 3, 0, 3, 'Uma malha feita de fios de metal, resistente e duravel.', 100, 'Malha');

INSERT INTO Armadura (idItem, defesa, defesaMagica, agilidade, descricao, valor, nome)
VALUES (3, 5, 0, 5, 'Uma armadura de aço bem equilibrada e confiável.', 200, 'Aço');

INSERT INTO Armadura (idItem, defesa, defesaMagica, agilidade, descricao, valor, nome)
VALUES (4, 0, 10, 0, 'Uma armadura mágica que confere poderes sobrenaturais ao seu portador.', 300, 'Mágico');

INSERT INTO Armadura (idItem, defesa, defesaMagica, agilidade, descricao, valor, nome)
VALUES (5, 10, 10, 10, 'Uma armadura mítica de origem divina, confere habilidades e poderes sobrenaturais.', 500, 'Mítico');
```

Entidade Ataque:

```
INSERT INTO Ataque (tipoAtaque, descricao, idHabilidade, danoFisico, danoMagico) VALUES (1, 'Ataque básico', 1, 10, 0);

INSERT INTO Ataque (tipoAtaque, descricao, idHabilidade, danoFisico, danoMagico) VALUES (2, 'Bola de fogo', 2, 0, 15);

INSERT INTO Ataque (tipoAtaque, descricao, idHabilidade, danoFisico, danoMagico) VALUES (3, 'Corte de espada', 3, 20, 0);

INSERT INTO Ataque (tipoAtaque, descricao, idHabilidade, danoFisico, danoMagico) VALUES (4, 'Mordida venenosa', 4, 10, 5);
```

```
INSERT INTO Ataque (tipoAtaque, descricao, idHabilidade, danoFisico, danoMagico) VALUES (5, 'Grito ensurdecador', 5, 0, 10);
```

Entidade Item:

```
INSERT INTO Item (idItem, tipo, idInventario, lojista) VALUES ('Armadura de couro', 1, 1);

INSERT INTO Item (idItem, tipo, idInventario, lojista) VALUES ('Espada de aço', 2, 1);

INSERT INTO Item (idItem, tipo, idInventario, lojista) VALUES ('Poção azul', 3, 1);

INSERT INTO Item (idItem, tipo, idInventario, lojista) VALUES ('Chave da porta enferrujada', 4, 1);

INSERT INTO Item (idItem, tipo, idInventario, lojista) VALUES ('Livro de feitiços', 5, 1);

INSERT INTO Item (idItem, tipo, idInventario, lojista) VALUES ('Amuleto da proteção', 6, 1);

INSERT INTO Item (idItem, tipo, idInventario, lojista) VALUES ('Anel de ataque', 7, 1);

INSERT INTO Item (idItem, tipo, idInventario, lojista) VALUES ('Poção de invisibilidade', 8, 1);

INSERT INTO Item (idItem, tipo, idInventario, lojista) VALUES ('LEG01', 'Legível', 9, 1);

INSERT INTO Item (idItem, tipo, idInventario, lojista) VALUES ('CONSU01', 'Chave da porta secreta', 10, 1);
```

Entidade Arma:

```
INSERT INTO Arma (idItem, ataqueFisico, ataqueMagico, descrição, valor, nome) VALUES (1, 10, 0, 'Espada de aço', 100, 'Espada');
INSERT INTO Arma (idItem, ataqueFisico, ataqueMagico, descrição, valor, nome) VALUES (2, 15, 0, 'Machado de batalha', 150, 'Machado');
INSERT INTO Arma (idItem, ataqueFisico, ataqueMagico, descrição, valor, nome) VALUES (3, 20, 0, 'Lança longa de Aço', 200, 'Lança');
INSERT INTO Arma (idItem, ataqueFisico, ataqueMagico, descrição, valor, nome) VALUES (4, 0, 10, 'Bisturi médico', 100, 'Bisturi');
```

Entidade Acessório:

```
INSERT INTO Acessorio (idItem, defesa, defesaMagica, agilidade, ataqueFisico, ataqueMagico, descricao, valor, nome) VALUES ('ACES01', 0, 15, 0, 0, 0, 'Um colar com uma pedra da alma acoplada. Protege contra os poderes sobrenaturais', NULL, 'Colar de pedra da Alma');

INSERT INTO Acessorio (idItem, defesa, defesaMagica, agilidade, ataqueFisico, ataqueMagico, descricao, valor, nome) VALUES ('ACES02', 0, 0, 0, 0, 5, 'Um anel com um diamante de sangue acoplado. Cheira a morte.', NULL, 'anel dos espectros');

INSERT INTO Acessorio (idItem, defesa, defesaMagica, agilidade, ataqueFisico, ataqueMagico, descricao, valor, nome) VALUES ('ACES03', 0, 0, 6, 0, 0, 'Um amuleto em formato de ratazana. As ratazanas deformadas parecem se interessar por ele e você se sente mais ágil quando o tem por perto', NULL, 'Amuleto de ratazana');

INSERT INTO Acessorio (idItem, defesa, defesaMagica, agilidade, ataqueFisico, ataqueMagico, descricao, valor, nome) VALUES ('ACES04', 0, 0, 0, 5, 0, 'Um óculos de grau. Enxergar melhor pode melhorar seus ataques.', NULL, 'Óculos');
```

Entidade Legível:


```

INSERT INTO
    Legivel (nome, conteúdo, descrição, valor)
VALUES
    (
        'O Livro dos Antigos',
        'O Livro dos Antigos é um texto sagrado que contém conhecimento sobre os deuses e a história do mundo.',
        'Este livro é um dos mais importantes do mundo de Fear and Hunger. Ele contém informações sobre os deuses, os ancestrais e a história do mundo. O livro é escrito em uma linguagem antiga e é muito difícil de entender.',
        5000
    );

INSERT INTO
    Legivel (nome, conteúdo, descrição, valor)
VALUES
    (
        'O Livro dos Monstros',
        'O Livro dos Monstros é uma enciclopédia de criaturas sobrenaturais que vivem no mundo de Fear and Hunger.',
        'Este livro é uma referência essencial para qualquer aventureiro que deseja aprender sobre os perigos que o aguardam. O livro contém informações sobre centenas de criaturas, incluindo seus poderes, fraquezas e como combatê-las.',
        2000
    );

INSERT INTO
    Legivel (nome, conteúdo, descrição, valor)
VALUES
    (
        'O Livro das Runas',
        'O Livro das Runas é um grimório que contém informações sobre magia e feitiçaria.',
        'Este livro é um tesouro de conhecimento para qualquer mago ou feiticeiro. O livro contém informações sobre uma variedade de magias, incluindo feitiços, encantamentos e runas.',
        3000
    );

INSERT INTO
    Legivel (nome, conteúdo, descrição, valor)
VALUES
    (
        'O Livro dos Segredos',
        'O Livro dos Segredos é um compêndio de conhecimentos ocultos e proibidos.',
        'Este livro é um item perigoso que deve ser manejado com cuidado. O livro contém informações sobre uma variedade de tópicos, incluindo magia negra, necromancia e ocultismo.',
        5000
    );

```

Entidade Consumível:

```

INSERT INTO Consumivel (idItem, nome, adHp, adMente, descrição, valor)
VALUES (1, 'Carne Crua', 20, 0, 'Um pedaço de carne crua. Pode ser consumido para restaurar a saúde, mas aumenta a chance de ficar doente.', 10);

INSERT INTO Consumivel (idItem, nome, adHp, adMente, descrição, valor)
VALUES (2, 'Carne Cozida', 40, 0, 'Um pedaço de carne cozida. Pode ser consumido para restaurar a saúde e a mente.', 20);

INSERT INTO Consumivel (idItem, nome, adHp, adMente, descrição, valor)
VALUES (3, 'Pão', 10, 0, 'Um pedaço de pão. Pode ser consumido para restaurar a saúde, mas não restaura a mente.', 5);

INSERT INTO Consumivel (idItem, nome, adHp, adMente, descrição, valor)

```

```
VALUES (4, 'Vinho', 0, 10, 'Uma garrafa de vinho. Pode ser consumida para restaurar a mente, mas não restaura a saúde.', 10);
```

Entidade Inventário:

```
INSERT INTO Inventario (idPersonagem, idItem)
VALUES (1, 1);
```

```
INSERT INTO Inventario (idPersonagem, idItem)
VALUES (2, 2);
```

Entidade Habilidade:

```
INSERT INTO Habilidade (idHabilidade, nome, custo, alma)
VALUES (1, 'Ataque Básico', 0, 'Feral');
```

```
INSERT INTO Habilidade (idHabilidade, nome, custo, alma)
VALUES (2, 'Bola de Fogo', 10, 'Caótica');
```

```
INSERT INTO Habilidade (idHabilidade, nome, custo, alma)
VALUES (3, 'Cura', 25, 'Liberdade');
```

Entidade Genital:

```
INSERT INTO Genital(idParte, Personagem, tipoParte, hpMaximo, hitChance, hpAtual) VALUES('G01', 'NI04', 'G', 30, 90, 30);
```

```
INSERT INTO Genital(idParte, Personagem, tipoParte, hpMaximo, hitChance, hpAtual) VALUES('G02', 'NI06', 'G', 25, 90, 40);
```

Entidade Cabeça:

```
INSERT INTO Cabeça(idParte, Personagem, tipoParte, hpMaximo, hitChance, hpAtual) VALUES ('C01', 'NI01', 'C', 5, 5, 5);
```

```
INSERT INTO Cabeça(idParte, Personagem, tipoParte, hpMaximo, hitChance, hpAtual) VALUES ('C02', 'NI02', 'C', 5, 5, 5);
```

```
INSERT INTO Cabeça(idParte, Personagem, tipoParte, hpMaximo, hitChance, hpAtual) VALUES ('C03', 'NI03', 'C', 5, 5, 5);
```

```
INSERT INTO Cabeça(idParte, Personagem, tipoParte, hpMaximo, hitChance, hpAtual) VALUES ('C04', 'NI04', 'C', 5, 5, 5);
```

```
INSERT INTO Cabeça(idParte, Personagem, tipoParte, hpMaximo, hitChance, hpAtual) VALUES ('C05', 'NI05', 'C', 5, 5, 5);
```

```
INSERT INTO Cabeça(idParte, Personagem, tipoParte, hpMaximo, hitChance, hpAtual) VALUES ('C06', 'NI06', 'C', 5, 5, 5);
```

Entidade Olho:

```
INSERT INTO Olho(idParte, Personagem, tipoParte, hpMaximo, hitChance, hpAtual) VALUES ('O01', 'NI01', 'O', 1, 85, 1);
```

```

INSERT INTO Olho(idParte, Personagem, tipoParte, hpMaximo, hitChance, hpAtual) VALUES ('002', 'NI02', 'O',
1, 85, 1,);

INSERT INTO Olho(idParte, Personagem, tipoParte, hpMaximo, hitChance, hpAtual) VALUES ('003', 'NI03', 'O',
1, 85, 1,);

INSERT INTO Olho(idParte, Personagem, tipoParte, hpMaximo, hitChance, hpAtual) VALUES ('004', 'NI04', 'O',
1, 85, 1,);

INSERT INTO Olho(idParte, Personagem, tipoParte, hpMaximo, hitChance, hpAtual) VALUES ('005', 'NI05', 'O',
1, 85, 1,);

INSERT INTO Olho(idParte, Personagem, tipoParte, hpMaximo, hitChance, hpAtual) VALUES ('006', 'NI06', 'O',
1, 85, 1,);

```

Entidade Braço:

```

INSERT INTO Braco(idParte, Personagem, tipoParte, hpMaximo, hitChance, hpAtual) VALUES ('B01', 'NI01', 'B',
20, 80, 20);

INSERT INTO Braco(idParte, Personagem, tipoParte, hpMaximo, hitChance, hpAtual) VALUES ('B02', 'NI01', 'B',
20, 80, 20);

INSERT INTO Braco(idParte, Personagem, tipoParte, hpMaximo, hitChance, hpAtual) VALUES ('B03', 'NI02', 'B',
10, 80, 10);

INSERT INTO Braco(idParte, Personagem, tipoParte, hpMaximo, hitChance, hpAtual) VALUES ('B04', 'NI02', 'B',
10, 80, 10);

```

Entidade Torço:

```

INSERT INTO Torco(idParte, Personagem, tipoParte, hpMaximo, hitChance, hpAtual) VALUES('T01', 'NI01', 'T',
40, 95, 40);

INSERT INTO Torco(idParte, Personagem, tipoParte, hpMaximo, hitChance, hpAtual) VALUES('T01', 'NI02', 'T',
30, 95, 30);

INSERT INTO Torco(idParte, Personagem, tipoParte, hpMaximo, hitChance, hpAtual) VALUES('T01', 'NI03', 'T',
90, 95, 90);

INSERT INTO Torco(idParte, Personagem, tipoParte, hpMaximo, hitChance, hpAtual) VALUES('T01', 'NI04', 'T',
150, 95, 150);

```

Entidade Perna:

```

INSERT INTO Perna(idParte, Personagem, tipoParte, hpMaximo, hitChance, hpAtual) VALUES('P01', 'NI01', 'P',
30, 85, 30);

INSERT INTO Perna(idParte, Personagem, tipoParte, hpMaximo, hitChance, hpAtual) VALUES('P02', 'NI01', 'P',
30, 85, 30);

INSERT INTO Perna(idParte, Personagem, tipoParte, hpMaximo, hitChance, hpAtual) VALUES('P03', 'NI02', 'P',
15, 85, 15);

INSERT INTO Perna(idParte, Personagem, tipoParte, hpMaximo, hitChance, hpAtual) VALUES('P04', 'NI02', 'P',
15, 85, 15);

```

Consultas

Consultar detalhes de um personagem específico:

```
SELECT * FROM Personagem WHERE idPersonagem = [ID do personagem];
```

Obter itens no inventário de um personagem:

```
SELECT I.*  
FROM Inventario AS Inv  
JOIN Item AS I ON Inv.FK_item = I.idItem  
WHERE Inv.FK_idPersonagem = [ID do personagem];
```

Obter habilidades associadas a uma alma específica:

```
SELECT H.*  
FROM Alma AS A  
JOIN Habilidade AS H ON A.idAlma = H.FK_alma  
WHERE A.nome = '[Nome da Alma]';
```

Consultar inimigos com defesa acima de um certo valor:

```
SELECT * FROM Inimigo WHERE defesa > [Valor de defesa];
```

Obter todos os itens de uma determinada lista:

```
SELECT I.*  
FROM Lista AS L  
JOIN Item AS I ON L.FK_item = I.idItem  
WHERE L.idLista = [ID da lista];
```

Consultar todos os personagens em um local específico:

```
SELECT P.*  
FROM Personagem AS P  
JOIN Local AS L ON P.FK2_local = L.idLocal  
WHERE L.idLocal = '[ID do Local]';
```

Obter detalhes da arma associada a um personagem:

```
SELECT Ar.*  
FROM PersonagemNaoJogavel AS PNJ  
JOIN Arma AS Ar ON PNJ.FK4_arma = Ar.idItem  
WHERE PNJ.FK_idPersonagem = [ID do personagem];
```

Consultar todas as almas coletadas por um personagem específico:

```
SELECT A.*
FROM Coletadas AS C
JOIN Alma AS A ON C.FK2_alma = A.idAlma
WHERE C.FK1_personagem = [ID do personagem];
```

Consultar todas as habilidades de um personagem baseadas em suas almas coletadas:

```
SELECT H.*
FROM Personagem AS P
JOIN Coletadas AS C ON P.idPersonagem = C.FK1_personagem
JOIN Alma AS A ON C.FK2_alma = A.idAlma
JOIN Habilidade AS H ON A.idAlma = H.FK_alma
WHERE P.idPersonagem = [ID do personagem];
```

Obter armadura associada a um personagem:

```
SELECT Am.*
FROM PersonagemNaoJogavel AS PNJ
JOIN Armadura AS Am ON PNJ.FK5_armadura = Am.idItem
WHERE PNJ.FK_idPersonagem = [ID do personagem];
```

Consultar inimigos com ataque mágico acima de um certo valor:

```
SELECT * FROM Inimigo WHERE ataqueMagico > [Valor de ataque mágico];
```

Obter todos os acessórios associados a um personagem não jogável:

```
-- Para o primeiro acessório:
SELECT Ac.*
FROM PersonagemNaoJogavel AS PNJ
JOIN Acessorio AS Ac ON PNJ.FK2_acessorio1 = Ac.idItem
WHERE PNJ.FK_idPersonagem = [ID do personagem];

-- Para o segundo acessório:
SELECT Ac.*
FROM PersonagemNaoJogavel AS PNJ
JOIN Acessorio AS Ac ON PNJ.FK3_acessorio2 = Ac.idItem
WHERE PNJ.FK_idPersonagem = [ID do personagem];
```

Consultar todos os itens de defesa mágica acima de um certo valor em uma lista:

```
SELECT I.*
FROM Lista AS L
JOIN Item AS I ON L.FK_item = I.idItem
WHERE L.idLista = [ID da lista] AND I.defesaMagica > [Valor de defesa mágica];
```

Obter detalhes de todos os personagens que possuem uma alma específica:

```
SELECT P.*
FROM Personagem AS P
JOIN Coletadas AS C ON P.idPersonagem = C.FK1_personagem
JOIN Alma AS A ON C.FK2_alma = A.idAlma
WHERE A.nome = '[Nome da Alma]';
```

Comprar itens na loja:

```
-- Deduzir o custo do item das moedas/recursos do jogador
-- (Assumindo que a tabela Personagem possui um campo 'moedas' para este exemplo)
UPDATE Personagem
SET moedas = moedas - (
    SELECT preco
    FROM Item
    WHERE idItem = [ID do item]
)
WHERE idPersonagem = [ID do personagem];

-- Adicionar o item ao inventário do jogador
INSERT INTO Inventario (FK_idPersonagem, FK_item)
VALUES ([ID do personagem], [ID do item]);
```

Consultar a saúde atual de uma determinada parte do corpo de um personagem:

```
SELECT tipoParte, hpAtual
FROM [Nome da Tabela da Parte do Corpo]
WHERE FK_personagem = [ID do personagem];
```

Obter todos os inimigos em um local específico:

```
SELECT I.*
FROM Inimigo AS I
JOIN Local AS L ON I.FK_local = L.idLocal
WHERE L.nome = '[Nome do Local]';
```

Normalização

Primeira Forma Normal

Descrição	Atributos devem ser monovalorados
Resolução 1	Cria-se uma nova relação entre a chave da relação original e o atributo multivalorado. Depois retira o atributo da relação original.
Resolução 2	Promove a quebra do atributo, um atributo que recebe dois parâmetros é especificado em dois atributos distintos

Análise:

1. Tabela de Estoque garante a primeira forma normal da relação entre Lojista e Item.
2. Tabela Coletadas garante a primeira forma normal da relação entre Personagem Jogável e Almas.
3. A quebra do atributo acessório em acessório1 e acessório2 garante a primeira forma normal.
4. Tabela Loot garante a primeira forma normal da relação entre Personagem e Local.

Segunda Forma Normal

Descrição	Não pode haver dependência parcial (onde um atributo depende apenas de parte da chave)
Resolução 1	A parte da chave e o atributo que depende desta formam outra relação. Logo depois, retiramos o atributo da relação original

Análise:

1. Analisando relações que possam existir dependência funcional parcial, no caso de relações com mais de uma chave e atributos ordinários, selecionamos estas relações: **Cabeça, Genital, Olho, Braço, Torço, Perna**. Nesses casos não há qualquer tipo de dependência entre os atributos comuns e as chaves parciais devido ao contexto em que se inserem essas tabelas. Sendo assim, não foi necessário fazer normalização, pois não se aplica nesse caso.

Segunda Forma Normal

Descrição	Não pode haver dependência transitiva de atributos comuns com qualquer superchave
Resolução 1	Cria-se uma nova relação na qual há atributos dependentes transitivamente e os atributos a quais dependem diretamente respectivamente. O atributo com a dependência transitiva sai da relação

Análise:

1. Não observamos nenhuma relação de dependência entre atributos, então, segundo o que diz a terceira forma normal, acreditamos que não seja necessária nenhuma alteração. Assim, o modelo está na terceira forma normal.

Forma Normal Boyce-Codd

Descrição	Para toda dependência funcional $X \rightarrow A$, X deve ser chave candidata
Resolução 1	Consideramos as dependências funcionais em que o lado esquerdo não é chave candidata e transformamos cada uma dessas dependências funcionais em novas relações. Logo, retiramos os correspondentes atributos - lado direito das DFs - da relação original
Observação	A Forma Normal de Boyce-Codd quando as formas normais primeira, segunda e terceira já estiverem sido verificadas é bastante rara

Análise:

1. Em todas as relações observadas, os atributos ordinários são dados isolados e portanto não possuem dependência entre eles e não há a presença de dependência parcial. Sendo assim, todas as dependências funcionais $X \rightarrow A$, X são chaves candidatas.

Quarta Forma Normal

Descrição	Não pode haver dependência multivalorada, pois causa redundância
Resolução	Em dependências multivaloradas de forma que A determina de forma multivalorada B, $A \twoheadrightarrow B$. Dividir a relação R em dois atributos, sendo A U B e R - B. Dessa forma não haverão mais dependências multivaloradas

Análise:

1. Assim como nos outros casos, não há relação de dependência entre atributos. Sendo assim, não houve necessidade de qualquer alteração. Sendo assim, o modelo está na quarta forma normal.