



Autarquia Educacional do Vale do São Francisco
Faculdade de Ciências Aplicadas e Sociais de Petrolina
Curso de Ciência da Computação

BANCO DE DADOS PARA LEAGUE OF LEGENDS
BANCO DE DADOS 2

PEDRO HENRIQUE CAMPOS DOS SANTOS

WAGNER LUCAS ARAUJO MIRANDA

Professor: RAPHAEL RIBEIRO VIDAL

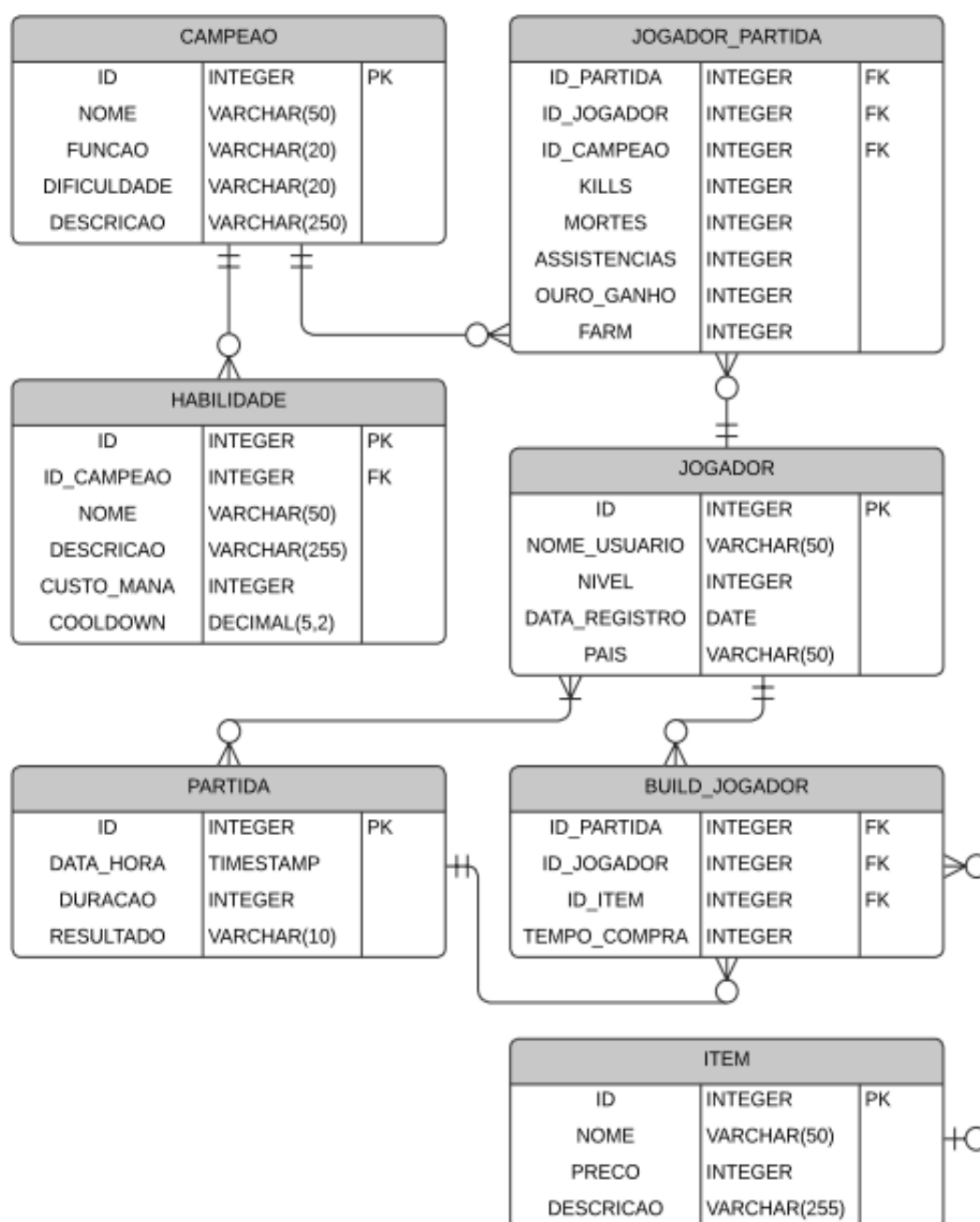
15/11/2024

SUMÁRIO

1.	DIAGRAMA	4
2.	PROJETO FÍSICO	5
2.1.	CRIAÇÃO DE DOMÍNIOS	5
2.1.1.	Domínio para representar o tipo de função de campeão	5
2.1.2.	Domínio para representar o resultado da partida	5
2.1.3.	Domínio para representar o nível do jogador	5
2.1.4.	Domínio para representar o custo de itens ou habilidades	5
2.1.5.	Domínio para representar nomes no sistema	5
2.2.	CRIAÇÃO DE TABELAS	5
2.2.1.	Tabela Campeão	5
2.2.2.	Tabela Habilidade	6
2.2.3.	Tabela Jogador	6
2.2.4.	Tabela Partida	7
2.2.5.	Tabela JogadorPartida	7
2.2.6.	Tabela Item	8
2.2.7.	Tabela BuildJogador	8
2.3.	ALTERAÇÃO DE TABELAS	8
2.3.1.	Alterar o tipo de dado do campo FUNCAO na tabela CAMPEAO	8
2.3.2.	Adicionar uma nova coluna REGIAO na tabela JOGADOR	9
2.3.3.	Alterar o nome do campo DESCRICAO para DETALHES na tabela ITEM	9
2.3.4.	Adicionar uma chave estrangeira na tabela PARTIDA para relacionar com REGIAO	9
2.3.5.	Remover a coluna TEMPO_COMPRA da tabela BUILD_JOGADOR	9
2.4.	INSERÇÃO DE DADOS NAS TABELAS	9
2.4.1.	Inserir um novo campeão na tabela CAMPEAO	9
2.4.2.	Inserir uma habilidade na tabela HABILIDADE	9
2.4.3.	Inserir um jogador na tabela JOGADOR	9
2.4.4.	Inserir uma partida na tabela PARTIDA	9
2.4.5.	Inserir um item na tabela ITEM	9
2.5.	ATUALIZAÇÃO DE DADOS DAS TABELAS	10
2.5.1.	Atualizar o preço de um item na tabela ITEM	10

2.5.2.	Alterar a função de um campeão na tabela CAMPEAO	10
2.5.3.	Atualizar o nível de um jogador na tabela JOGADOR	10
2.5.4.	Corrigir a duração de uma partida na tabela PARTIDA	10
2.5.5.	Alterar a descrição de uma habilidade na tabela HABILIDADE	10
2.6.	EXCLUSÃO DE DADOS DAS TABELAS	11
2.6.1.	Excluir um campeao da tabela CAMPEAO	11
2.6.2.	Excluir um item da tabela ITEM	11
2.6.3.	Excluir um jogador da tabela JOGADOR	11
2.6.4.	Excluir uma habilidade específica da tabela HABILIDADE	11
2.6.5.	Excluir uma partida da tabela PARTIDA	11
2.7.	VISÕES	11
2.7.1.	Visão com os nomes dos campeões	11
2.7.2.	Visão com os nomes dos jogadores	11
2.7.3.	Visão com os nomes dos itens	12
2.7.4.	Visão para listar as partidas com informações resumidas	12
2.7.5.	Visão para listar os itens mais utilizados em partidas	12
2.8.	PROCEDIMENTOS	13
2.8.1.	Procedimento para adicionar um novo campeão	13
2.8.2.	Procedimento para atualizar um nível de um jogador	13
2.8.3.	Procedimento para registrar uma partida	14
2.8.4.	Procedimento para remover um item do banco	14
2.9.	EXCEÇÕES	14
2.9.1.	Exceção para impedir a inserção de campeões sem nome	14
2.9.2.	Exceção para evitar níveis negativos para jogadores	15
2.9.3.	Exceção para impedir resultados inválidos em partidas	15
2.9.4.	Exceção para evitar a inserção de habilidade sem campeão associado	16
2.9.5.	Exceção para verificar valores negativos em preço de itens	16
2.10.	GATILHOS	16
2.10.1.	Gatilho para registrar a data e hora da última atualização do jogador	16
2.10.2.	Gatilho para calcular automaticamente o tempo total de partida	17
2.10.3.	Gatilho para evitar a duplicidade de builds por jogador	17
2.10.4.	Gatinha para registrar a data de inserção de um novo campeão	17
2.10.5.	Gatilho para impedir a exclusão de jogadores com nível alto	18

1. DIAGRAMA



2. PROJETO FÍSICO

2.1. CRIAÇÃO DE DOMÍNIOS

2.1.1. Domínio para representar o tipo de função de campeão

```
CREATE DOMAIN FUNC_TYPE AS VARCHAR(20)
```

```
CHECK (VALUE IN ('Assassino', 'Mago', 'Suporte', 'Tanque', 'Lutador',  
'Atirador'));
```

2.1.2. Domínio para representar o resultado da partida

```
CREATE DOMAIN RESULT_TYPE AS VARCHAR(10)
```

```
CHECK (VALUE IN ('Vitória', 'Derrota'));
```

2.1.3. Domínio para representar o nível do jogador

```
CREATE DOMAIN LEVEL_TYPE AS INTEGER
```

```
CHECK (VALUE >= 0 AND VALUE <= 18);
```

2.1.4. Domínio para representar o custo de itens ou habilidades

```
CREATE DOMAIN COST_TYPE AS INTEGER
```

```
CHECK (VALUE >= 0);
```

2.1.5. Domínio para representar nomes no sistema

```
CREATE DOMAIN NAME_TYPE AS VARCHAR(50)
```

```
NOT NULL;
```

2.2. CRIAÇÃO DE TABELAS

2.2.1. Tabela Campeão

```
CREATE TABLE CAMPEAO (
```

```
    ID INTEGER NOT NULL,
```

```
    NOME VARCHAR(50) NOT NULL,
```

```
    FUNCAO VARCHAR(20) NOT NULL,
```

```
DIFICULDADE VARCHAR(20),  
DESCRICAO VARCHAR(250),  
CONSTRAINT PK_CAMPEAO PRIMARY KEY (ID)  
);
```

2.2.2. Tabela Habilidade

```
CREATE TABLE HABILIDADE (  
    ID INTEGER NOT NULL,  
    ID_CAMPEAO INTEGER NOT NULL,  
    NOME VARCHAR(50) NOT NULL,  
    DESCRICAO VARCHAR(255),  
    CUSTO_MANA INTEGER CHECK (CUSTO_MANA >= 0),  
    COOLDOWN DECIMAL(5, 2) CHECK (COOLDOWN >= 0),  
    CONSTRAINT PK_HABILIDADE PRIMARY KEY (ID),  
    CONSTRAINT FK_HABILIDADE_CAMPEAO FOREIGN KEY  
    (ID_CAMPEAO) REFERENCES CAMPEAO(ID)  
);
```

2.2.3. Tabela Jogador

```
CREATE TABLE JOGADOR (  
    ID INTEGER NOT NULL,  
    NOME_USUARIO VARCHAR(50) NOT NULL,  
    NIVEL INTEGER CHECK (NIVEL >= 0 AND NIVEL <= 100),  
    DATA_REGISTRO DATE NOT NULL,  
    PAIS VARCHAR(50),  
    CONSTRAINT PK_JOGADOR PRIMARY KEY (ID)  
);
```

2.2.4. Tabela Partida

```
CREATE TABLE PARTIDA (  
    ID INTEGER NOT NULL,  
    DATA_HORA TIMESTAMP NOT NULL,  
    DURACAO INTEGER CHECK (DURACAO > 0),  
    RESULTADO VARCHAR(10) CHECK (RESULTADO IN ('Vitória',  
'Derrota')),  
    CONSTRAINT PK_PARTIDA PRIMARY KEY (ID)  
);
```

2.2.5. Tabela JogadorPartida

```
CREATE TABLE JOGADOR_PARTIDA (  
    ID_PARTIDA INTEGER NOT NULL,  
    ID_JOGADOR INTEGER NOT NULL,  
    ID_CAMPEAO INTEGER NOT NULL,  
    KILLS INTEGER CHECK (KILLS >= 0),  
    MORTES INTEGER CHECK (MORTES >= 0),  
    ASSISTENCIAS INTEGER CHECK (ASSISTENCIAS >= 0),  
    OURO_GANHO INTEGER CHECK (OURO_GANHO >= 0),  
    FARM INTEGER CHECK (FARM >= 0),  
    CONSTRAINT PK_JOGADOR_PARTIDA PRIMARY KEY  
(ID_PARTIDA, ID_JOGADOR, ID_CAMPEAO),  
    CONSTRAINT FK_JP_PARTIDA FOREIGN KEY (ID_PARTIDA)  
REFERENCES PARTIDA(ID),  
    CONSTRAINT FK_JP_JOGADOR FOREIGN KEY (ID_JOGADOR)  
REFERENCES JOGADOR(ID),  
    CONSTRAINT FK_JP_CAMPEAO FOREIGN KEY (ID_CAMPEAO)  
REFERENCES CAMPEAO(ID)
```

);

2.2.6. Tabela Item

```
CREATE TABLE ITEM (
    ID INTEGER NOT NULL,
    NOME VARCHAR(50) NOT NULL,
    PRECO INTEGER CHECK (PRECO >= 0),
    DESCRICAO VARCHAR(255),
    CONSTRAINT PK_ITEM PRIMARY KEY (ID)
);
```

2.2.7. Tabela BuildJogador

```
CREATE TABLE BUILD_JOGADOR (
    ID_PARTIDA INTEGER NOT NULL,
    ID_JOGADOR INTEGER NOT NULL,
    ID_ITEM INTEGER NOT NULL,
    TEMPO_COMPRA INTEGER CHECK (TEMPO_COMPRA >= 0),
    CONSTRAINT PK_BUILD_JOGADOR PRIMARY KEY (ID_PARTIDA,
    ID_JOGADOR, ID_ITEM),
    CONSTRAINT FK_BJ_PARTIDA FOREIGN KEY (ID_PARTIDA)
    REFERENCES PARTIDA(ID),
    CONSTRAINT FK_BJ_JOGADOR FOREIGN KEY (ID_JOGADOR)
    REFERENCES JOGADOR(ID),
    CONSTRAINT FK_BJ_ITEM FOREIGN KEY (ID_ITEM)
    REFERENCES ITEM(ID)
);
```


2.3. ALTERAÇÃO DE TABELAS

- 2.3.1. Alterar o tipo de dado do campo FUNCAO na tabela CAMPEAO**
ALTER TABLE CAMPEAO ALTER FUNCAO TYPE VARCHAR(30);
- 2.3.2. Adicionar uma nova coluna REGIAO na tabela JOGADOR**
ALTER TABLE JOGADOR ADD REGIAO VARCHAR(50);
- 2.3.3. Alterar o nome do campo DESCRICAO para DETALHES na tabela ITEM**
ALTER TABLE ITEM ALTER COLUMN DESCRICAO TO DETALHES;
- 2.3.4. Adicionar uma chave estrangeira na tabela PARTIDA para relacionar com REGIAO**
ALTER TABLE PARTIDA ADD ID_REGIAO INTEGER;

ALTER TABLE PARTIDA ADD CONSTRAINT FK_PARTIDA_REGIAO
FOREIGN KEY (ID_REGIAO) REFERENCES JOGADOR(ID);
- 2.3.5. Remover a coluna TEMPO_COMPRA da tabela BUILD_JOGADOR**
ALTER TABLE BUILD_JOGADOR DROP TEMPO_COMPRA;

2.4. INSERÇÃO DE DADOS NAS TABELAS

- 2.4.1. Inserir um novo campeão na tabela CAMPEAO**
INSERT INTO CAMPEAO (ID, NOME, FUNCAO, DIFICULDADE, DESCRICAO)

VALUES (1, 'Ahri', 'Mago', 'Média', 'Ahri é uma maga ágil com habilidades de controle de multidão e dano explosivo.');
- 2.4.2. Inserir uma habilidade na tabela HABILIDADE**
INSERT INTO HABILIDADE (ID, ID_CAMPEAO, NOME, DESCRICAO, CUSTO_MANA, COOLDOWN)

VALUES (1, 1, 'Orbe da Ilusão', 'Ahri lança um orbe que causa dano na ida e na volta.', 65, 7.5);
- 2.4.3. Inserir um jogador na tabela JOGADOR**
INSERT INTO JOGADOR (ID, NOME_USUARIO, NIVEL, DATA_REGISTRO, PAIS)

VALUES (1, 'Raphel', 35, '2024-11-16', 'Brasil');
- 2.4.4. Inserir uma partida na tabela PARTIDA**
INSERT INTO PARTIDA (ID, DATA_HORA, DURACAO, RESULTADO)

VALUES (1, '2024-11-15 20:30:00', 35, 'Vitória');

2.4.5. Inserir um item na tabela ITEM

```
INSERT INTO ITEM (ID, NOME, PRECO, DESCRICAO)
```

```
VALUES (1, 'Espada Longa', 350, 'Um item básico que  
aumenta o dano de ataque.');
```

2.5. ATUALIZAÇÃO DE DADOS DAS TABELAS**2.5.1. Atualizar o preço de um item na tabela ITEM**

```
UPDATE ITEM
```

```
SET PRECO = 400
```

```
WHERE NOME = 'Espada Longa';
```

2.5.2. Alterar a função de um campeão na tabela CAMPEAO

```
UPDATE CAMPEAO
```

```
SET FUNCAO = 'Assassino'
```

```
WHERE NOME = 'Ahri';
```

2.5.3. Atualizar o nível de um jogador na tabela JOGADOR

```
UPDATE JOGADOR
```

```
SET NIVEL = 36
```

```
WHERE ID = 1;
```

2.5.4. Corrigir a duração de uma partida na tabela PARTIDA

```
UPDATE PARTIDA
```

```
SET DURACAO = 40
```

```
WHERE ID = 1;
```

2.5.5. Alterar a descrição de uma habilidade na tabela HABILIDADE

```
UPDATE HABILIDADE
```

```
SET DESCRICAO = 'Ahri lança um orbe mágico que causa dano na ida e  
dano verdadeiro na volta.'
```

```
WHERE NOME = 'Orbe da Ilusão';
```

2.6. EXCLUSÃO DE DADOS DAS TABELAS

2.6.1. Excluir um campeão da tabela CAMPEAO

```
DELETE FROM CAMPEAO
```

```
WHERE NOME = 'Ahri';
```

2.6.2. Excluir um item da tabela ITEM

```
DELETE FROM ITEM
```

```
WHERE NOME = 'Espada Longa';
```

2.6.3. Excluir um jogador da tabela JOGADOR

```
DELETE FROM JOGADOR
```

```
WHERE ID = 1;
```

2.6.4. Excluir uma habilidade específica da tabela HABILIDADE

```
DELETE FROM HABILIDADE
```

```
WHERE NOME = 'Orbe da Ilusão';
```

2.6.5. Excluir uma partida da tabela PARTIDA

```
DELETE FROM PARTIDA
```

```
WHERE ID = 1;
```

2.7. VISÕES

2.7.1. Visão com os nomes dos campeões

```
CREATE VIEW VIEW_NOME_CAMPEOES AS
```

```
SELECT
```

```
    ID,
```

```
    NOME
```

```
FROM CAMPEAO;
```

2.7.2. Visão com os nomes dos jogadores

```
CREATE VIEW VIEW_NOME_JOGADORES AS
```

```
SELECT
```

```
ID,  
NOME_USUARIO  
FROM JOGADOR;
```

2.7.3. Visão com os nomes dos itens

```
CREATE VIEW VIEW_NOME_ITENS AS  
  
SELECT  
  
ID,  
  
NOME  
  
FROM ITEM;
```

2.7.4. Visão para listar as partidas com informações resumidas

```
CREATE VIEW VIEW_PARTIDAS AS  
  
SELECT  
  
P.ID AS ID_PARTIDA,  
  
P.DATA_HORA,  
  
P.DURACAO,  
  
P.RESULTADO  
  
FROM PARTIDA P;
```

2.7.5. Visão para listar os itens mais utilizados em partidas

```
CREATE VIEW VIEW_ITENS_UTILIZADOS AS  
  
SELECT  
  
B.ID_ITEM,  
  
I.NOME AS NOME_ITEM,  
  
COUNT(B.ID_ITEM) AS VEZES_UTILIZADO  
  
FROM BUILD_JOGADOR B  
  
INNER JOIN ITEM I ON B.ID_ITEM = I.ID
```

```
GROUP BY B.ID_ITEM, I.NOME;
```

2.8. Procedimentos

2.8.1. Procedimento para adicionar um novo campeão

```
CREATE PROCEDURE ADD_CAMPEAO (
    NOME_CAMPEAO VARCHAR(50),
    FUNCAO_CAMPEAO VARCHAR(20),
    DIFICULDADE_CAMPEAO VARCHAR(20),
    DESCRICAO_CAMPEAO VARCHAR(250)
)
AS
BEGIN
    INSERT INTO CAMPEAO (NOME, FUNCAO, DIFICULDADE,
DESCRICAO)
    VALUES (:NOME_CAMPEAO, :FUNCAO_CAMPEAO,
:DIFICULDADE_CAMPEAO, :DESCRICAO_CAMPEAO);
END;
```

2.8.2. Procedimento para atualizar o nível de um jogador

```
CREATE PROCEDURE ATUALIZAR_NIVEL_JOGADOR (
    ID_JOGADOR_PARAM INTEGER,
    NOVO_NIVEL INTEGER
)
AS
BEGIN
    UPDATE JOGADOR
    SET NIVEL = :NOVO_NIVEL
    WHERE ID = :ID_JOGADOR_PARAM;
```

END;

2.8.3. Procedimento para registrar uma nova partida

```
CREATE PROCEDURE REGISTRAR_PARTIDA (
    DATA_HORA_PARTIDA TIMESTAMP,
    DURACAO_PARTIDA INTEGER,
    RESULTADO_PARTIDA VARCHAR(10)
)
AS
BEGIN
    INSERT INTO PARTIDA (DATA_HORA, DURACAO, RESULTADO)
    VALUES (:DATA_HORA_PARTIDA, :DURACAO_PARTIDA,
:RESULTADO_PARTIDA);
END;
```

2.8.4. Procedimento para remover um item do banco

```
CREATE PROCEDURE REMOVER_ITEM (
    ID_ITEM_PARAM INTEGER
)
AS
BEGIN
    DELETE FROM ITEM
    WHERE ID = :ID_ITEM_PARAM;
END;
```

2.9. Exceções

2.9.1. Exceção para impedir a inserção de campeões sem nome

```
CREATE TRIGGER TRG_CAMPEAO_SEM_NOME
BEFORE INSERT ON CAMPEAO
AS
```

```

BEGIN
    IF (NEW.NOME IS NULL OR TRIM(NEW.NOME) = '') THEN
        BEGIN
            EXCEPTION "Erro: O nome do campeão não pode estar vazio.";
        END
    END;

```

2.9.2. Exceção para evitar níveis negativos para jogadores

```

CREATE TRIGGER TRG_NIVEL_NEGATIVO
BEFORE UPDATE OR INSERT ON JOGADOR
AS
BEGIN
    IF (NEW.NIVEL < 0) THEN
        BEGIN
            EXCEPTION "Erro: O nível do jogador não pode ser negativo.";
        END
    END;

```

2.9.3. Exceção para impedir resultados inválidos em partidas

```

CREATE TRIGGER TRG_RESULTADO_INVALIDO
BEFORE INSERT OR UPDATE ON PARTIDA
AS
BEGIN
    IF (NEW.RESULTADO NOT IN ('VITÓRIA', 'DERROTA', 'EMPATE'))
    THEN
        BEGIN
            EXCEPTION "Erro: O resultado da partida deve ser 'VITÓRIA',
            'DERROTA' ou 'EMPATE'.";
        END
    END;

```

END;

2.9.4. Exceção para evitar a inserção de habilidades sem campeão associado

CREATE TRIGGER TRG_HABILIDADE_SEM_CAMPEAO

BEFORE INSERT OR UPDATE ON HABILIDADE

AS

BEGIN

IF (NEW.ID_CAMPEAO IS NULL OR NOT EXISTS (SELECT 1 FROM CAMPEAO WHERE ID = NEW.ID_CAMPEAO)) THEN

BEGIN

EXCEPTION "Erro: A habilidade deve estar associada a um campeão existente.";

END

END;

2.9.5. Exceção para verificar valores negativos em preço de itens

CREATE TRIGGER TRG_PRECO_NEGATIVO

BEFORE INSERT OR UPDATE ON ITEM

AS

BEGIN

IF (NEW.PRECO < 0) THEN

BEGIN

EXCEPTION "Erro: O preço do item não pode ser negativo.";

END

END;

2.10. Gatilhos

2.10.1. Gatilho para registrar a data e hora da última atualização do jogador

CREATE TRIGGER TRG_ATUALIZA_DATA_JOGADOR

BEFORE UPDATE ON JOGADOR

AS

BEGIN

 NEW.DATA_REGISTRO = CURRENT_DATE; -- Atualiza a data de registro para a data atual

END;

2.10.2. Gatilho para calcular automaticamente o tempo total de partida

CREATE TRIGGER TRG_CALCULA_TEMPO_PARTIDA

AFTER INSERT ON PARTIDA

AS

BEGIN

 UPDATE PARTIDA

 SET DURACAO = EXTRACT(MINUTE FROM NEW.DATA_HORA + INTERVAL '30' MINUTE); -- Adiciona 30 minutos ao registro inicial como exemplo

END;

2.10.3. Gatilho para evitar duplicidade de builds por jogador

CREATE TRIGGER TRG_PREVENIR_BUILD_DUPLICADA

BEFORE INSERT ON BUILD_JOGADOR

AS

BEGIN

 IF (EXISTS (SELECT 1 FROM BUILD_JOGADOR
 WHERE ID_JOGADOR = NEW.ID_JOGADOR
 AND ID_ITEM = NEW.ID_ITEM)) THEN

 BEGIN

 EXCEPTION "Erro: O jogador já possui esse item na build.";

 END

END;

2.10.4. Gatilho para registrar a data de inserção de um novo campeão

```
CREATE TRIGGER TRG_DATA_INSERTAO_CAMPEAO
```

```
BEFORE INSERT ON CAMPEAO
```

```
AS
```

```
BEGIN
```

```
    NEW.DATA_INSERTAO = CURRENT_DATE;
```

```
END;
```

2.10.5. Gatilho para impedir a exclusão de jogadores com nível alto

```
CREATE TRIGGER TRG_IMPEDIR_EXCLUSAO_JOGADOR
```

```
BEFORE DELETE ON JOGADOR
```

```
AS
```

```
BEGIN
```

```
    IF (OLD.NIVEL > 30) THEN
```

```
        BEGIN
```

```
            EXCEPTION "Erro: Jogadores com nível acima de 30 não podem  
ser excluídos.";
```

```
        END
```

```
END;
```