**Banco de Dados**

**Prova 3**

**Alunos: João Pedro Azevedo Veras dos Reis; Luís Guilherme Freitas de Almeida Silva**

**Matrículas: 2020034378; 2020002820**

**Modelo Relacional**

**Diagrama

Descrição gerada automaticamente**

**Modelo Lógico**

Personagem(cod\_personagem, cod\_pirata, cod\_marinha)

PK (cod\_personagem)

FK (cod\_pirata) referencia PIRATA (cod\_pirata)

FK (cod\_marinha) referencia MARINHA (cod\_marinha)

Pirata (cod\_pirata, nome, recompensa, cod\_ilha, cod\_tripulacao)

PK (cod\_pirata)

FK (cod\_ilha) referencia ILHA (cod\_ilha)

FK (cod\_tripulacao) referencia TRIPULACAO (cod\_tripulacao)

Marinha (cod\_marinha, nome, recompensa, cod\_ilha, cod\_tripulacao)

PK (cod\_marinha)

FK (cod\_ilha) referencia ILHA (cod\_ilha)

FK (cod\_tripulacao) referencia TRIPULACAO (cod\_tripulacao)

Arma (cod\_arma, nome)

PK (cod\_arma)

AkumaNoMi (cod\_fruta, nome, cod\_tipo, cod\_personagem)

PK (cod\_fruta)

FK (cod\_tipo) referencia TIPO (cod\_tipo)

FK (cod\_personagem) referencia PERSONAGEM (cod\_personagem)

Tipo (cod\_tipo, nome)

PK (cod\_tipo)

Ilha (cod\_ilha, nome, cod\_arco)

PK (cod\_ilha)

FK (cod\_arco) referencia ARCO (cod\_arco)

Arco (cod\_arco, nome)

PK (cod\_arco)

Tripulacao (cod\_tripulacao, nome, cod\_alianca)

PK (cod\_tripulacao)

FK (cod\_alianca) referencia ALIANCA (cod\_alianca)

Alianca (cod\_alianca, nome)

PK (cod\_alianca)

ArmaPersonagem (cod\_armapersonagem, cod\_personagem, cod\_arma)

PK (cod\_armapersonagem)

FK (cod\_personagem) referencia PERSONAGEM (cod\_personagem)

FK (cod\_arma) referencia ARMA (cod\_arma)

**Modelo Físico**

CREATE TABLE ALIANCA (

cod\_alianca serial,

nome varchar,

primary key(cod\_alianca)

);

CREATE TABLE ARCO (

cod\_arco serial,

nome varchar,

primary key(cod\_arco)

);

CREATE TABLE TIPO (

cod\_tipo serial,

nome varchar,

primary key(cod\_tipo)

);

CREATE TABLE TRIPULACAO (

cod\_tripulacao serial,

nome varchar,

cod\_alianca integer,

primary key(cod\_tripulacao),

foreign key(cod\_alianca) references ALIANCA(cod\_alianca)

);

CREATE TABLE ILHA (

cod\_ilha serial,

nome varchar,

cod\_arco integer,

primary key(cod\_ilha),

foreign key(cod\_arco) references ARCO(cod\_arco)

);

CREATE TABLE PIRATA (

cod\_pirata serial,

nome varchar,

recompensa float,

cod\_ilha integer,

cod\_tripulacao integer,

primary key (cod\_pirata),

foreign key(cod\_ilha) references ILHA (cod\_ilha),

foreign key(cod\_tripulacao) references TRIPULACAO (cod\_tripulacao)

);

CREATE TABLE MARINHA (

cod\_marinha serial,

nome varchar,

recompensa float,

cod\_ilha integer,

cod\_tripulacao integer,

primary key (cod\_marinha),

foreign key(cod\_ilha) references ILHA (cod\_ilha),

foreign key(cod\_tripulacao) references TRIPULACAO (cod\_tripulacao)

);

CREATE TABLE PERSONAGEM (

cod\_personagem serial,

cod\_pirata integer,

cod\_marinha integer,

primary key (cod\_personagem),

foreign key(cod\_pirata) references PIRATA (cod\_pirata),

foreign key(cod\_marinha) references MARINHA (cod\_marinha)

);

CREATE TABLE ARMA (

cod\_arma serial,

nome varchar,

primary key (cod\_arma)

);

CREATE TABLE AKUMANOMI (

cod\_fruta serial,

nome varchar,

cod\_tipo integer,

cod\_personagem integer,

primary key(cod\_fruta),

foreign key (cod\_tipo) references TIPO(cod\_tipo),

foreign key (cod\_personagem) references PERSONAGEM(cod\_personagem)

);

CREATE TABLE ARMAPERSONAGEM (

cod\_armapersonagem serial,

cod\_personagem integer,

cod\_arma integer,

primary key (cod\_armapersonagem),

foreign key (cod\_personagem) references PERSONAGEM(cod\_personagem),

foreign key (cod\_arma) references ARMA (cod\_arma)

);

**FUNCTION E TRIGGER**

-- FUNCTION E TRIGGER

CREATE TABLE PIRATARECOMPENSAS (

codigo serial,

cod\_pirata integer,

data\_alteracao date,

recompensa float,

primary key (codigo),

foreign key (cod\_pirata) references PIRATA(cod\_pirata)

);

/\*Adicona o valor da recompensa nova na tabela PIRATA RECOMPENSA

assim que o valor for atualizado ou inserido na tabela PIRATA acionando a TRIGGER

e consequentemente a FUNCTION\*/

CREATE OR REPLACE FUNCTION atualizarecompensa() returns TRIGGER as $$

BEGIN

INSERT INTO PIRATARECOMPENSAS (cod\_pirata, data\_alteracao, recompensa)

VALUES (new.cod\_pirata, NOW(), new.recompensa);

RETURN NULL;

END;

$$ LANGUAGE PLPGSQL;

CREATE TRIGGER tg\_autaliza\_recompensa

AFTER INSERT OR UPDATE

ON PIRATA

FOR EACH ROW

EXECUTE FUNCTION atualizarecompensa();

**Inserts Básicos**

INSERT INTO tipo VALUES(1, 'Paramecia');

INSERT INTO tipo VALUES(2, 'Logia');

INSERT INTO tipo VALUES(3, 'Zoan');

INSERT INTO tipo VALUES(4, 'Zoan Mitica');

INSERT INTO pirata VALUES(0, null, null, 0, 0);

INSERT INTO marinha VALUES(0, null, null, 0, 0);

INSERT INTO ilha VALUES(0, null, null);

INSERT INTO tripulacao VALUES(0, null, 0);

INSERT INTO alianca VALUES (0, null);

INSERT INTO arco VALUES (0, null);

**Consultas**

-- CONSULTAS EXISTS E HAVING

SELECT p.nome FROM PIRATA p

WHERE EXISTS(

SELECT cod\_tripulacao FROM TRIPULACAO T

WHERE t.cod\_tripulacao = p.cod\_tripulacao

AND lower(t.nome) = 'piratas do chapéu de palha');

SELECT cod\_pirata FROM PIRATARECOMPENSAS

GROUP BY cod\_pirata

HAVING SUM(recompensa) > 300000;