**Tabelas de Inserção e Queries - Banco de Dados de League of Legends**

**Alunos: Leonardo Alves, Lucas Gustavo Cavalcanti e Raphael Garcia**

**Feito no POSTGRESQL**

**TABELAS DE INSERÇÂO**

create TABLE emote(

nome varchar(20) CONSTRAINT pk\_nome\_emote PRIMARY key,

preco INTEGER not NULL

);

create TABLE elo(

nome VARCHAR(12) CONSTRAINT pk\_nome\_elo PRIMARY key

);

create TABLE temporada(

ano INTEGER CONSTRAINT pk\_ano\_temporada PRIMARY key

);

create TABLE servidor(

nome varchar(3) CONSTRAINT pk\_nome\_servidor PRIMARY key,

num\_jogadores INTEGER not NULL,

ano INTEGER not NULL

);

create TABLE classe(

nome VARCHAR(10) CONSTRAINT pk\_nome\_classe PRIMARY key

);

create TABLE colecao\_skin(

nome varchar(20) CONSTRAINT pk\_nome\_colecao PRIMARY key,

ano INTEGER not NULL

);

create TABLE raridade(

tipo varchar(8) CONSTRAINT pk\_tipo\_raridade PRIMARY key,

valor INTEGER not NULL

);

create TABLE jogador(

apelido varchar(40) CONSTRAINT pk\_apelido\_jogador PRIMARY key,

num\_vit INTEGER not NULL,

num\_der INTEGER not NULL,

ama FLOAT not NULL,

nivel\_honra INTEGER not NULL,

nivel\_jog INTEGER not NULL,

nome\_elo varchar(12) not NULL,

pdl INTEGER not NULL,

FOREIGN KEY (nome\_elo)

REFERENCES elo(nome)

);

CREATE TABLE amizade(

apelido\_jog1 varchar(40) ,

apelido\_jog2 VARCHAR(40) not NULL,

data\_amizade DATE not NULL,

FOREIGN key(apelido\_jog1)

REFERENCES jogador(apelido),

FOREIGN key(apelido\_jog2)

REFERENCES jogador(apelido)

);

ALTER TABLE amizade ADD CONSTRAINT pk\_apelidos\_amizade PRIMARY KEY(apelido\_jog1,apelido\_jog2);

CREATE TABLE partida(

id serial CONSTRAINT pk\_id\_partida PRIMARY key,

data\_part DATE not NULL,

duracao INTEGER not NULL,

mapa VARCHAR(17) not NULL,

nome\_serv VARCHAR(3) not NULL,

FOREIGN key(nome\_serv)

REFERENCES servidor(nome)

);

CREATE TABLE recompensa(

titulo VARCHAR(22) CONSTRAINT pk\_titulo\_recompensa PRIMARY KEY,

ano\_recom INTEGER,

FOREIGN key(ano\_recom)

REFERENCES temporada(ano)

);

CREATE TABLE campeao(

nome varchar(15) CONSTRAINT pk\_nome\_camp PRIMARY key,

regiao varchar(17) not NULL,

ano\_lan INTEGER not NULL,

classeP VARCHAR(10) not NULL,

classeS VARCHAR(10),

preco INTEGER not NULL,

FOREIGN key(classeP)

REFERENCES classe(nome),

FOREIGN key(classeS)

REFERENCES classe(nome)

);

CREATE TABLE atualizacao(

numero serial CONSTRAINT pk\_num\_atual PRIMARY KEY,

nota varchar(13) not NULL,

ano\_at INTEGER not NULL,

FOREIGN key(ano\_at)

REFERENCES temporada(ano)

);

CREATE TABLE skin\_campeao(

nome varchar(40) CONSTRAINT pk\_nome\_skin\_campeao PRIMARY key,

ano\_lanc INTEGER not NULL,

legado BOOLEAN not NULL,

nome\_colec VARCHAR(20),

tipo\_rar VARCHAR(8) not NULL,

nome\_camp VARCHAR(15) not NULL,

FOREIGN key(nome\_colec)

REFERENCES colecao\_skin(nome),

FOREIGN key(tipo\_rar)

REFERENCES raridade(tipo),

FOREIGN key (nome\_camp)

REFERENCES campeao(nome)

);

CREATE TABLE croma(

nome varchar(13) ,

nome\_skin VARCHAR(40) not NULL,

FOREIGN key (nome\_skin)

REFERENCES skin\_campeao(nome)

);

ALTER TABLE croma ADD CONSTRAINT pk\_nomeCorSkin\_croma PRIMARY KEY(nome,nome\_skin);

CREATE TABLE escolha(

apelido\_jog varchar(40) CONSTRAINT pk\_escolha\_apelido PRIMARY key,

nome\_camp VARCHAR(15) not NULL,

id serial not NULL,

funcao varchar(10),

FOREIGN key(id)

REFERENCES partida(id),

FOREIGN key(apelido\_jog)

REFERENCES jogador(apelido),

FOREIGN key(nome\_camp)

REFERENCES campeao(nome)

);

alter table escolha drop COLUMN apelido\_jog;

alter TABLE escolha add COLUMN apelido\_jog VARCHAR(40) not NULL;

ALTER TABLE escolha add FOREIGN key(apelido\_jog) REFERENCES jogador(apelido);

alter table escolha add CONSTRAINT pks\_apelido\_nome PRIMARY key(apelido\_jog,id);

CREATE TABLE ranqueada(

id serial CONSTRAINT pk\_id\_ranqueada PRIMARY key,

ano\_temp INTEGER not NULL,

mmr INTEGER not NULL,

FOREIGN key(id)

REFERENCES partida(id),

FOREIGN key(ano\_temp)

REFERENCES temporada(ano)

);

CREATE TABLE elo\_recompensa(

elo\_nome varchar(12) CONSTRAINT pk\_tier\_eloRecomp PRIMARY key,

titulo\_recomp VARCHAR(22) not NULL,

FOREIGN key(elo\_nome)

REFERENCES elo(nome),

FOREIGN key(titulo\_recomp)

REFERENCES recompensa(titulo)

);

alter table elo\_recompensa drop COLUMN elo\_nome;

alter TABLE elo\_recompensa add COLUMN elo\_nome VARCHAR(12) not NULL;

ALTER TABLE elo\_recompensa add FOREIGN key(elo\_nome) REFERENCES elo(nome);

alter table elo\_recompensa add CONSTRAINT pks\_elo\_titulo PRIMARY key(elo\_nome,titulo\_recomp);

CREATE TABLE elo\_temporada(

elo\_nome varchar(12) CONSTRAINT pk\_tier\_eloTemporada PRIMARY key,

ano\_temp INTEGER not NULL,

FOREIGN key(elo\_nome)

REFERENCES elo(nome),

FOREIGN key(ano\_temp)

REFERENCES temporada(ano)

);

alter table elo\_temporada drop COLUMN elo\_nome;

alter TABLE elo\_temporada add COLUMN elo\_nome VARCHAR(12) not NULL;

ALTER TABLE elo\_temporada add FOREIGN key(elo\_nome) REFERENCES elo(nome);

alter table elo\_temporada add CONSTRAINT pks\_elo\_ano PRIMARY key(elo\_nome,ano\_temp);

CREATE TABLE jog\_emote(

apelido\_jog varchar(40) CONSTRAINT pk\_jogEmote\_apelido PRIMARY key,

nome\_emote VARCHAR(20) not NULL,

FOREIGN key(apelido\_jog)

REFERENCES jogador(apelido),

FOREIGN key(nome\_emote)

REFERENCES emote(nome)

);

alter table jog\_emote drop COLUMN apelido\_jog;

alter TABLE jog\_emote add COLUMN apelido\_jog VARCHAR(40) not NULL;

ALTER TABLE jog\_emote add FOREIGN key(apelido\_jog) REFERENCES jogador(apelido);

alter table jog\_emote add CONSTRAINT pks\_apelido\_nomeEmote PRIMARY key(apelido\_jog,nome\_emote);

CREATE TABLE jog\_campeao(

apelido\_jogador VARCHAR(40),

FOREIGN KEY (apelido\_jogador) REFERENCES jogador(apelido),

nome\_campeao VARCHAR (15),

FOREIGN key (nome\_campeao) REFERENCES campeao(nome)

);

ALTER TABLE jog\_campeao add CONSTRAINT pks\_apelidoj\_nomec PRIMARY key(apelido\_jogador, nome\_campeao);

CREATE TABLE jog\_skin(

apelido\_jogador VARCHAR(40),

FOREIGN KEY (apelido\_jogador) REFERENCES jogador(apelido),

nome\_skin VARCHAR (40),

FOREIGN key (nome\_skin) REFERENCES skin\_campeao(nome)

);

ALTER TABLE jog\_skin add CONSTRAINT pks\_apelidoj\_nomesk PRIMARY key(apelido\_jogador, nome\_skin);

**QUERIES**

**I)** **Nome, Ano de Lançamento e Classe Primária de Campeões que não possuem Classe Secundária**

SELECT nome, ano\_lan, classep FROM campeao WHERE classes IS NULL

**II) ID e Duração de Partidas que Ocorreram no Servidor NA**

SELECT id, duracao FROM partida WHERE nome\_serv='NA'

**III) ID de Partidas que tiveram o Campeão Lulu**

SELECT id FROM partida WHERE id in (SELECT id from escolha WHERE nome\_camp = 'Lulu')

**IV) Nome e Ano de Lançamento de Skins de Campeão cuja Coleção de Skin foi lançada depois de 2019**

SELECT nome FROM skin\_campeao WHERE nome\_colec in (SELECT nome FROM colecao\_skin WHERE 2019<ano)

**V) Apelido e A/M/A de Jogadores de Elo igual a Diamante e de A/M/A acima de 3**

SELECT apelido, ama FROM jogador WHERE nome\_elo='Diamante' and ama>3

**VI) Nome e Raridade das Skins de Campeão de Raridade Épica, possuídas por mais de 3 Jogadores**

SELECT DISTINCT nome\_skin, tipo\_rar FROM (jog\_skin FULL OUTER JOIN skin\_campeao ON nome\_skin=nome) WHERE tipo\_rar='Épico' AND nome\_skin IN (SELECT nome\_skin FROM jog\_skin GROUP BY nome\_skin HAVING COUNT(\*)>3)

**VII) Total de Jogadores que possuem as Skins de Campeões que são dadas como Recompensa (Skins Vitoriosas)**

SELECT COUNT(apelido\_jogador) from jog\_skin where nome\_skin in (SELECT nome FROM (skin\_campeao INNER JOIN recompensa ON nome=titulo))

**VIII) Total de Skins de Campeão que fazem parte de uma Coleção de Skin**

SELECT COUNT(nome) from skin\_campeao where nome\_colec is not NULL;

**IX) Média do PDL de Jogadores abaixo do Elo Mestre**

SELECT CAST((SELECT AVG(pdl) from jogador WHERE nome\_elo != 'Desafiante' and nome\_elo !='Grão-Mestre' and nome\_elo !='Mestre') as NUMERIC(10,2));

**X) Apelido dos Jogadores presentes na Partida de maior duração no Servidor BR, e a duração da Partida**

SELECT apelido\_jog,duracao from (escolha right JOIN partida on escolha.id=partida.id) where duracao = (SELECT MAX(duracao) from partida WHERE nome\_serv = 'BR');