Projeto CC2005

parte 2

Grupo 77

Dante Rotnes - 202105772

Porto, Novembro 2022

GitHub

Este arquivo e todos os outros arquivos utilizados nesse projeto estão disponíveis no seguinte repositório no GitHub:

<https://github.com/dRotnes/Base-de-Dados>

Correção do modelo da BD

Correção de requisitos

Para resolver os problemas encontrados no primeiro relatório, realizamos as seguintes mudan;as no modelo da BD?

1. A entidade **Tecnico** agora possui alguns atributos a mais

2. A entidade **Jogo** agora possui os atributos **PontosCasa** e **PontosVisitante**

3. Os relacionamentos redundantes time-**treinado por**-tecnico e time-**possui**-estadio foram retirados

4. O grau dos relacionamentos foi alterado para representarem corretamente as relações:

* **Time - joga fora - Jogo (1:N)**
* **Time - joga em casa - Jogo (1:N)**
* **Técnico - treina - Time (1:1)**
* **Jogador - joga para - Time (N:1)**
* **Estádio - pertence a - Time (1:N)**
* **Estádio - recebe - Jogo (1:N)**

Isso nos leva à necessidade de uma correção nos Modelos ER e Relacional que agora seguem as relações destacadas acima.





Povoamento das Tabelas

Para efetuar o povoamento das tabelas, usamos o excel que havia referenciado no primeiro relatório e que se encontra disponibilizado no github. Os dados no ficheiro excel estão atualizados até o dia 19/12/2022. Todos os dados encontrados no ficheiro excel foram retirados dos seguintes websites:

Lista de arenas: <https://en.hispanosnba.com/nba-arenas>

Lista de times:

<https://geojango.com/pages/list-of-nba-teams>

Calendário dos jogos: <https://fixturedownload.com/sport/basketball>

Informação acerca dos jogadores: <https://www.nba.com/teams>

Informação acerca dos técnicos: <http://www.espn.com/nba/coaches>

Outras informações: <https://www.nba.com/stats>

Os dados foram então organizados para seguir o esquema da nossa base de dados.

Para inserir os dados na tabelas, foi utilizado o comando LOAD DATA INFILE fazendo referência aos CSVs de cada tabela (disponibilizados no github). Exemplo:

Como podemos ver, foi necessário para a tabela **Jogos**, especificar como seria o tipo de data a ser recebida.

Para cada tabela foram feitos os seguintes números de entradas:

|  |  |
| --- | --- |
| **Jogos** | **1230** |
| **Jogadores** | **500** |
| **Tecnicos** | **30** |
| **Estadios** | **29** |
| **Times** | **30** |

Aplicação Python

A aplicação python, seguindo o modelo apresentado em aula, possui 16 endpoints:

|  |  |
| --- | --- |
| **ENDPOINT** | **FUNCIONALIDADE** |
| / | Página de entrada com barras de pesquisas para caada tabela (menos a de jogos) |
| /teams | Lista de todos os times |
| /teams/<id> | Lista atributos específicos de um time |
| /teams/<id>/games | Lista todos os jogos de um time específico |
| /teams/search/<expr> | Lista todos os times que possuem o input passado no nome |
| /players | Lista todos os jogadores |
| /players/<id> | Lista atributos específicos de um jogador |
| /players/search/<expr> | Lista todos os jogadores que possuem o input passado no nome |
| /coaches | Lista todos os tecnicos |
| /coaches/<id> | Lista atributos específicos de um tecnico |
| /coaches/search/<expr> | Lista todos os tecnicos que possuem o input passado no nome |
| /stadiums | Lista todos os estadios |
| /stadiums/<id> | Lista atributos específicos de um estadio |
| /stadiums/search/<expr> | Lista todos os estadios que possuem o input passado no nome |
| /games | Lista todos os jogos |
| /games/<id> | Lista atributos específicos de um jogo |

Para cada endpoint há diferentes queries sqls. Seguem as queries que representam o ponto 3:

OBS.: Em alguns casos utilizamos uma maneira de inserir o parâmetro apenas uma vez, rodando db.execute("SET @id:=%s;", id) ao início da função

|  |  |
| --- | --- |
| **Descrição** | **Query** |
| Query para buscar últimos 10 jogos de um time:  1º busca todos os jogos onde o idTimeCasa é o id passado e pontos difere de 0 (significa que já foi jogado), cruzando com as tabelas Teams e Estadios para buscar nome dos times e estadio onde o jogo ocorre.  2º busca todos os jogos onde o idTimeFora é o id passado e pontos difere de 0 (significa que já foi jogado), cruzando com as tabelas Teams e Estadios para buscar nome dos times e estadio onde o jogo ocorre.  3º Seleciona a união de ambas as queries anteriores e ordena por dataHora decrescente (para trazer do mais próximo pro mais distante) | with cte as (          Select j.id, j.dataHora, j.pontosCasa as pontos, j.pontosVisitante as pontosAgainst, j.idEstadio, t1.nome as team , t2.nome as against, t2.id as idAgainst, e.nome as estadio          from Jogos j inner join Teams t1 on j.idTeamCasa = t1.id          inner join Teams t2 on j.idTeamFora = t2.id          inner join Estadios e on j.idEstadio = e.id          where t1.id = @id and j.pontosCasa <> 0        ),        cte2 as(          Select j.id, j.dataHora, j.pontosVisitante as pontos, j.pontosCasa as pontosAgainst, j.idEstadio, t1.nome as team, t2.nome as against, t2.id as idAgainst, e.nome as estadio          from Jogos j inner join Teams t1 on j.idTeamFora = t1.id          inner join Teams t2 on j.idTeamCasa = t2.id          inner join Estadios e on j.idEstadio = e.id          where t1.id = @id and j.pontosVisitante <> 0        )        Select id,dataHora, pontos, pontosAgainst, team, against, idEstadio, idAgainst, estadio from cte        union all        Select id,dataHora, pontos, pontosAgainst, team, against, idEstadio, idAgainst, estadio  from cte2        order by dataHora DESC limit 10; |
| Query para encontrar um jogador que possua um certo input no nome.  utiliza um concat para poder utiizar o like como ‘%input%’, retornando então todos os jogadores que possuírem em algum lugar do nome o dado input | SELECT id, nome      FROM Jogadores      WHERE nome LIKE CONCAT('%%', %s, '%%'); |
| Query para encontrar os próximos jogos que serão jogados num certo estádio  cruza com a tabela Teams duas vezes, para buscar o nome de ambos os times da casa e visitante e, busca pelos jogos que não tiveram pontos e foram realizados no estadio especificado | Select j.id, j.dataHora , t1.nome as teamCasa , t2.nome as teamFora, t1.id as idCasa, t2.id as idFora      from Jogos j inner join Teams t1 on j.idTeamCasa = t1.id      inner join Teams t2 on j.idTeamFora = t2.id      where j.idEstadio = %s and j.pontosCasa = 0 |