

**UNIVERSIDADE FEDERAL DE MATO GROSSO - UFMT
CAMPUS ARAGUAIA**

TAINÁ ISABELA MONTEIRO DA SILVA

**BARRA DO GARÇAS
2022**

Tainá Isabela Monteiro Da Silva

Trabalho escrito apresentado como requisito parcial
para obtenção de nota na disciplina de Banco de
Dados em Bacharel em Ciência da Computação do
Universidade Federal de Mato Grosso - UFMT,
Campus Araguaia.

Docente: Prof. Dr. Linder Cândido da Silva

Barra do Garças
2022

SUMÁRIO

1	INTRODUÇÃO	4
2	DESENVOLVIMENTO	5
3	CONSULTAS SOLICITADAS	7
4	RESTRIÇÃO DE INTEGRIDADE	12
5	RESTRIÇÃO DE INTEGRIDADE REFERENCIAL	13
6	INSERÇÃO SEM VIOLAÇÃO DE RESTRIÇÃO	14

1 Introdução

Para produzir o banco de dados, foi criado um arquivo SQL para originar as tabelas e suas respectivas colunas, o banco foi nomeado como copa.sql, todo o banco foi criado com base nas informações disponibilizadas pelo docente.

2 Desenvolvimento

Para a primeira etapa no processo de criação deste banco de dados foi criado o arquivo SQL (copa.sql) contendo a criação do banco nomeado por copa, e a criação de todas as cinco tabelas sendo elas:

- pais
- jogadores
- resultados_jogos
- cartoes_jogadores
- gols_assistencias_jogadores

Para a segunda etapa foi utilizado os recursos disponibilizados pelo próprio MySQL workbench para inserção de dados csv na tabela selecionada:

```
Import Results

File /home/taina/Documentos/UFMT/BD/Trabalho BD Copa - Tainá Isabela/Dados/Pais.csv was imported in 0.128 s
Table copa.pais has been used
32 records imported
```

Após fazer todas as importações de dados csv para as respectivas tabelas, de forma que respeitassem as dependências e chaves estrangeiras entre si, realizei uma seleção total para confirmar devidamente se os dados foram importados corretamente, logo abaixo segue o log do MySql Workbench para as seleções, mostrando que todas as tabelas foram preenchidas corretamente:

51	23:35:32	SHOW SESSION VARIABLES LIKE 'lower_case_table_names'	OK	0,000 sec
52	23:35:32	SHOW DATABASES	OK	0,000 sec
53	23:35:33	SHOW SESSION VARIABLES LIKE 'lower_case_table_names'	OK	0,000 sec
54	23:35:33	SHOW COLUMNS FROM `copa`.'jogadores'	OK	0,000 sec
55	23:35:35	PREPARE stmt FROM 'INSERT INTO `copa`.'jogadores` (...)'	OK	0,000 sec
56	23:35:39	DEALLOCATE PREPARE stmt	OK	0,000 sec
57	23:35:59	SHOW SESSION VARIABLES LIKE 'lower_case_table_names'	OK	0,000 sec
58	23:35:59	SHOW DATABASES	OK	0,000 sec
59	23:35:59	SHOW SESSION VARIABLES LIKE 'lower_case_table_names'	OK	0,000 sec
60	23:35:59	SHOW COLUMNS FROM `copa`.'gols_assistencias_joga...'	OK	0,000 sec
61	23:36:01	PREPARE stmt FROM 'INSERT INTO `copa`.'gols_assiste...'	OK	0,000 sec
62	23:36:02	DEALLOCATE PREPARE stmt	OK	0,000 sec
63	23:36:39	SHOW SESSION VARIABLES LIKE 'lower_case_table_names'	OK	0,000 sec
64	23:36:39	SHOW DATABASES	OK	0,000 sec
65	23:36:40	SHOW SESSION VARIABLES LIKE 'lower_case_table_names'	OK	0,000 sec
66	23:36:40	SHOW COLUMNS FROM `copa`.'cartoes_jogadores'	OK	0,000 sec
67	23:36:41	PREPARE stmt FROM 'INSERT INTO `copa`.'cartoes_joga...'	OK	0,000 sec
68	23:36:42	DEALLOCATE PREPARE stmt	OK	0,000 sec
69	23:37:38	select * from cartoes_jogadores LIMIT 0, 500	164 row(s) returned	0,00061 sec / 0,000...
70	23:37:38	select * from gols_assistencias_jogadores LIMIT 0, 500	168 row(s) returned	0,00063 sec / 0,000...
71	23:37:38	select * from jogadores LIMIT 0, 500	500 row(s) returned	0,00071 sec / 0,000...
72	23:37:38	select * from pais LIMIT 0, 500	32 row(s) returned	0,00056 sec / 0,000...

Action: DEALLOCATE PREPARE stmt
 Response: OK 0,000 sec
 Duration: 0,000 sec 0,000 sec

3 Consultas Solicitadas

Esta parte tem como intuito explicar a lógica por trás das escolhas de respostas para as consultas requerida pelo trabalho, sendo que os resultados obtidos através destas estão organizados em uma pasta organizada por enunciado.

1 - Recupere o nome, a posição e o clube dos jogadores do país ‘USA’.

Para esta questão é necessário apenas a utilização da cláusula where para filtrar o país por "USA" dentro da tabela de jogadores:

```
select nome, posicao, clube from jogadores where pais="USA";
```

2 - Recupere os nomes dos países participantes da copa do mundo de 2014 que ganharam a copa do mundo pelo menos uma vez.

Novamente, basta a utilização de uma cláusula where para filtrar os países participantes atuais que já participaram de alguma copa e ganharam:

```
select nome_do_pais from pais where numero_de_vitorias_em_copas>=1;
```

3 - Recupere os nomes dos países participantes da copa do mundo de 2014 que nunca ganharam uma copa do mundo.

A estrutura se mantém igual a ultima consulta porém com uma pequena mudança no where:

```
select nome_do_pais from pais where numero_de_vitorias_em_copas=0;
```

4 - Recupere o nome e o país do jogador com o maior número de cartões amarelos na copa do mundo de 2014.

Para recuperar o nome e o país do jogador esses dois dados não estão armazenados na mesma tabela, para isso, usamos a cláusula join, mais especificamente a inner join. Sendo a cláusula Inner Join correspondente a uma operação de junção em álgebra relacional, ou seja, com essa cláusula conseguimos combinar a tabela cartoes_jogadores com a tabela de jogadores, essa junção pode ser feita quando ambas as tabelas de interesse mantém dados que compõem nas duas, nesse caso utilizamos o id que aparece em ambas tabelas, após a junção utilizamos a where para filtrar apenas os jogadores com cartões amarelos pedido na questão:

```
select j.nome, j.pais from cartoes_jogadores inner join jogadores j on
cartoes_jogadores.jogador_id = j.jogador_id where numero_de_cartoes_amarelos = (select
max(numero_de_cartoes_amarelos) from cartoes_jogadores);
```

5 - Para cada cidade sede, recupere a cidade sede e o número total de partidas que nela foram disputadas.

Para isso é necessário utilizar a cláusula count, com essa função retornamos o número de linhas que compõem a coluna especificada da tabela. Além disso também é necessário utilizar a instrução group by, que serve para agrupar dados pela semelhança delas, o propósito de usá-la foi agrupar os dados em torno da coluna cidade_sede da tabela resultados_jogos.

```
select c.cidade_sede, count(c.cidade_sede) as numero_total_partidas from
resultados_jogos c group by c.cidade_sede;
```

6 - Para cada país, recupere o nome do país e o número de jogos que ele disputou como Time1 na tabela RESULTADOS_JOGOS, bem como o total de gols marcados (Soma de Gols_time1) e gols tomados (SUM de Gols_time2).

Como falado na questão anterior sobre o uso de agrupamento e os exemplos das funções que podem ser utilizadas com o group by, aqui recorremos ao uso do sum, a função sum

apresentada retorna a soma total de uma coluna do tipo numérico. Logo, dado o funcionamento que utilizamos desta para somar os gols dos times da copa da tabela de resultados_jogos, vale lembrar que só foi possível com a finalidade do uso de agrupamento do time1.

```
select time1 as nome_do_pais, count(t.time1) as numero_de_jogos_pelo_pais,
sum(t.gols_time1) as numero_gols_marcados_time1, sum(t.gols_time2) as
gols_tomados_do_time2 from resultados_jogos t group by t.time1;
```

7 - Para cada país, recupere o nome do país e o número de jogos que ele disputou como Time2 na tabela RESULTADOS_JOGOS, bem como o total de gols marcados (Soma de Gols_time2) e gols tomados (SUM de Gols_time1).

De modo apresentado na questão acima, a alternativa aqui foi alterar para o time 2 e agrupar usando o mesmo.

```
select time2 as nome_do_pais, count(t.time2) as numero_de_jogos_pelo_pais,
sum(t.gols_time2) as numero_gols_marcados_time2, sum(t.gols_time1) as
gols_tomados_do_time1 from resultados_jogos t group by t.time2;
```

8 - Escreva uma consulta para obter o número total de partidas que cada país disputou (seja como Time1 ou como Time2), o número total de gols marcados e o número total de gols tomados. A saída deve estar ordenada em ordem decrescente do número de jogos disputados.

Ao final dessa consulta obteremos o NomePaís, a soma do número de jogos apelidada por NoDeJogos, soma do total de gols marcados, apelidada também por TotalGolsMarcados e por fim o total de gols tomados, apelidada por TotalGolsTomados, com isso, utilizamos como estratégia o uso de da junção de duas tabelas criadas, primeiro criamos a tabela com o resultado referente ao time 1, como podemos observar o group by time1, logo após, fizemos o mesmo processo porém obtendo os dados do time 2, essa união foi feita pelo operador union que combina resultados de uma ou mais queries. Para finalizar, após a união de duas tabelas ficamos com uma “terceira” tabela que nesse caso será a junção das duas com os dados pedidos na questão, como a

soma de número de jogos ordenado por ordem decrescente.

```
select NomePaís, sum(NoDeJogos) as NoDeJogos, sum(TotalGolsMarcados) as TotalGolsMarcados, sum(TotalGolsTomados) as TotalGolsTomados from ( select time1 as NomePaís, count(*) as NoDeJogos, sum(gols_time1) as TotalGolsMarcados, sum(gols_time2) as TotalGolsTomados from resultados_jogos group by time1 union select time2 as NomePaís, count(*) as NoDeJogos, sum(gols_time2) as TotalGolsMarcados, sum(gols_time1) as TotalGolsTomados from resultados_jogos group by time2, group by NomePaís order by sum(NoDeJogos) desc );
```

9 - Liste todas as partidas disputadas pelo país ‘Brazil’ como Time1 ou Time2.

Para essa basta utilizar uma cláusula or (ou), ou seja, só referência mesmo a busca de primeiro time como Brazil ou como segundo time como Brazil, utilizamos assim conseguimos obter as partidas disputadas tanto como time 1 e como time 2 pelo Brazil.

```
select * from resultados_jogos where time1 = 'Brazil' or time2 = 'Brazil' ;
```

10 - Recupere os nomes dos jogadores que marcaram pelo menos um gol, o país desses jogadores, e o número de gols que cada um marcou. Ordene o resultado pelo número de gols marcados em ordem decrescente.

Visto as demais questões para essa utilizei o conhecimento juntado até então e recuperei os jogadores utilizando a união de tabelas com o inner join e filtrando com where, assim, foi a forma mais fácil de chegar a resposta e só por final utilizando o order by para ordenar os resultados.

```
select j.nome, j.pais, gols_assistencias_jogadores.gols as num_de_gols_ordem_desc from gols_assistencias_jogadores inner join jogadores j on gols_assistencias_jogadores.jogador_id = j.jogador_id where gols >= 1 order by gols desc;
```

11 - Repita a consulta 10, mas somente para os jogadores que tiveram mais de 2 gols.

A única diferença foi no where no qual filtro utilizou o operador de maior para pegar os resultados dos jogadores acima de dois gols.

```
select j.nome, j.pais, gols_assistencias_jogadores.gols as num_de_gols_ordem_desc from
    gols_assistencias_jogadores inner join jogadores j on
gols_assistencias_jogadores.jogador_id = j.jogador_id where gols > 2 order by gols desc;
```

12 - Liste os países participantes da copa de 2014 e a sua população, ordenando em ordem decrescente de população.

```
select nome_do_pais, populacao from pais order by populacao desc;
```

4 Restrição de Integridade

Primeira inserção: Inserindo, na tabela cartões_jogadores, uma tupla com jogador_id que não existe na tabela de jogadores. insert into cartoes_jogadores values (999999, 90, 09);

Segunda inserção: Inserindo um valor que já existe na tabela dos países. insert into pais values ('Brazil', 293.4 , 4, 'Pedro Henrique');

Terceira inserção: Inserção de uma nova tupla, com os cartões, amarelos e vermelhos, porém sem o identificador do jogador. O campo jogador_id da tabela em questão é not null. insert into cartoes_jogadores (numero_de_cartoes_amarelos, numero_de_cartoes_vermelhos) values (90, 85);

5 Restrição de Integridade Referencial

Para o erro de inserção de Restrição de Integridade Referencial foi feita uma tentativa de deletar a tupla com o nome do país igual a Brazil da tabela dos países. O que é que existe outra tabela que faz referência à tabela pais, sendo ela resultado_jogos. Caso tivesse sido definido a chave estrangeira para deletar junto (on delete cascade) o erro não aconteceria, já que tanto a tupla com o nome do país igual a Brazil quanto a tupla que faz referência a ela seriam deletadas.

```
delete from pais where nome_do_pais='Brazil';
```

6 Inserção sem Violação de Restrição

Cada inserção listada abaixo funcionou como esperado. Primeiro é inserido no banco mais um país para compor a lista de países da copa de 2014. A próxima inserção diz respeito a um dos jogadores da seleção de futebol masculino da Angola. Desconsiderando o ID do jogador que foi criado por nós, o restante dos dados é real, incluindo o país em questão. Por último o número de cartões vermelhos e amarelos referente a esse jogador recém adicionado. Como ele não participou da copa, já que o país Angola não esteve presente, decidimos colocar zero nos dois valores de cartão.

- insert into pais values ('Angola', 39.89, 0, 'Vinicius Camelo');
- insert into jogadores values (99999, 'MESSI FERNANDES', 'MESSI', 'FERNANDES', '1979-12-8', 'Angola', 188, 'Moreirense FC', 'Goleiro', 37, 0);
- insert into cartoes_jogadores values (99999, 0, 0);