

A Série A1 do Campeonato Paulista de Futebol de 2019, ou Paulistão Sicredi 2019, por motivos de patrocínio, foi a 118ª edição da principal divisão do futebol paulista. Foi realizada e organizada pela Federação Paulista de Futebol e disputada por 16 clubes entre os dias 19 de janeiro e 21 de abril.

#### **Regulamento:**

O campeonato foi disputado por dezesseis clubes divididos em quatro grupos. Na primeira fase, os times enfrentaram apenas os clubes dos outros grupos, totalizando doze rodadas. Os dois melhores classificados de cada chave avançaram às quartas-de-final. Todos os confrontos da fase eliminatória aconteceram em dois jogos e, em caso de empate em pontos (uma vitória para cada time ou dois empates), o primeiro critério de desempate foi o saldo de gols na fase final. Os dois times que somaram menos pontos na primeira fase foram rebaixados, independentemente dos grupos em que jogaram.

Também ficou decidida a volta do Troféu do Interior, em que os dois primeiros colocados ganharam o direito de disputar a Copa do Brasil de 2020. Os classificados teriam que ficar entre a 9ª e a 14ª colocação na tabela de classificação.

Além disso, os três clubes mais bem classificados que não pertençam a nenhuma divisão do Campeonato Brasileiro tiveram vaga na Série D de 2020. Caso entre esses estivessem as seguintes equipes: Botafogo-SP, Bragantino, Corinthians, Guarani, Oeste, Palmeiras, Ponte Preta, Santos, São Bento ou São Paulo, cada um deles estaria automaticamente fora da disputa, abrindo espaço para uma outra equipe. O campeonato adotará o VAR a partir das quartas-de-final.

#### **Atividade:**

Fazer uma aplicação em Java (Web (Servlets) ou Desktop (Swing ou JavaFX)) com SQL Server para resolver os problemas, da seguinte maneira: O sistema deve ter 3 tabelas principais:

- Times (Com todos os 20 times)(Não é necessário CRUD para ela)

Times (CodigoTime | NomeTime | Cidade | Estadio)

- Grupos (Corinthians, Palmeiras, Santos e São Paulo NÃO PODEM estar no mesmo grupo) (A coluna Grupo não pode aceitar nenhum valor diferente de A, B, C, D)

Grupos (Grupo | CodigoTime)

- Jogos (A primeira fase ocorrerá em 12 datas seguidas, sempre rodada cheia (os 8 jogos), aos domingos e quartas)

Jogos (CodigoTimeA | CodigoTimeB | GolsTimeA | GolsTimeB | Data)

O sistema deve se comportar da seguinte maneira:

Uma tela deve chamar uma procedure que divide os times nos quatro grupos, preenchendo, aleatoriamente (com exceção da regra já exposta em Grupos).

Uma tela deve gerar as rodadas dos jogos, de acordo com as regras do campeonato, preenchendo a tabela jogos.

Lembre-se, cada rodada tem 8 jogos (todos os 16 times). Lembre-se também que, as rodadas vão acontecer de quarta e domingo, sucessivamente, sem pausas.

Uma tela deve mostrar 4 Tabelas com os 4 grupos formados.

Uma tela deve mostrar um Campo, onde o usuário digite a data e, em caso de ser uma data com rodada, mostre uma tabela com todos os jogos daquela rodada.

- Um jogo não pode ocorrer 2 vezes, mesmo em rodadas diferentes
- Um time não pode aparecer 2 vezes na mesma rodada
- A fase de grupos vai terminar antes da data final do campeonato, uma vez que o campeonato prevê datas das fase eliminatórias também

Times Participantes:

▲ Equipes promovidas da Série A2 de 2018

Equipe	Cidade	Em 2018	Estádio	Material Esportivo	Títulos
Botafogo-SP	Ribeirão Preto	8º	Santa Cruz	Kappa	0
Bragantino	Bragança Paulista	6º	Nabi Abi Chedid	Icone Sports	1 (em 1990)
Corinthians	São Paulo	1º	Arena Corinthians	Nike	29 (último em 2018)
Ferroviária	Araraquara	13º	Fonte Luminosa	Lupo	0
Guarani	Campinas	1º ▲ (A2)	Brinco de Ouro da Princesa	Topper	0
Ituano	Itu	12º	Novelli Júnior	Kanxa	2 (último em 2014)
Mirassol	Mirassol	10º	José Maria de Campos Maia	Nakal	0
Novorizontino	Novo Horizonte	5º	Jorge Ismael de Biasi	Nakal	0
Oeste	Barueri	2º ▲ (A2)	Arena Barueri	Deka	0
Palmeiras	São Paulo	2º	Allianz Parque	Puma	22 (último em 2008)
Ponte Preta	Campinas	9º	Moisés Lucarelli	Topper	0
Red Bull Brasil	Campinas	14º	Moisés Lucarelli	Nike	0
Santos	Santos	4º	Vila Belmiro Pacaembu <sup>[a]</sup>	Umbro	22 (último em 2016)
São Bento	Sorocaba	11º	Walter Ribeiro	Joma	0
São Caetano	São Caetano do Sul	7º	Anacleto Campanella	Sanka Sports	1 (em 2004)
São Paulo	São Paulo	3º	Morumbi	Adidas	21 (último em 2005)

OBS: A. \* A Vila Belmiro estava fechada para a reforma até março de 2019, com isso o Santos mandou seus jogos no Estádio do Pacaembu.<sup>[7]</sup>

- Apesar de o Santos mandar jogos no Pacaembú à época, seu estádio deve ser considerado a Vila Belmiro

Para conhecimento, sempre é possível utilizar *Alias* em consultas tipo JOIN em SQL.

Normalmente, usamos *Alias* no nome da coluna (Ex.: SELECT GETDATE() AS 'Hoje'). O "AS" determina o *Alias*.

No entanto, é possível se usar *Alias* no nome da tabela também. De acordo com a documentação Microsoft:

A legibilidade de uma instrução SELECT pode ser aprimorada, atribuindo um alias a uma tabela, que também é conhecido como nome de correlação ou variável de intervalo. Um alias de tabela pode ser atribuído com ou sem a palavra-chave AS:

- *table\_name AS table alias*
- *table\_name table\_alias*

No exemplo a seguir, o alias *c* é atribuído a *Customer* e o alias *s* é atribuído a *Store*.

```
USE AdventureWorks2008R2;  
GO  
SELECT c.CustomerID, s.Name  
FROM Sales.Customer AS c  
JOIN Sales.Store AS s  
ON c.CustomerID = s.BusinessEntityID ;
```

Se um alias for atribuído a uma tabela, todas as referências explícitas à tabela na instrução do Transact-SQL precisarão usar o alias; não o nome de tabela. Por exemplo, a instrução SELECT a seguir gera um erro de sintaxe porque utiliza o nome da tabela quando existe um alias atribuído:

```
SELECT Sales.Customer.CustomerID, /* Illegal reference to  
Sales.Customer. */  
       s.Name  
FROM Sales.Customer AS c  
JOIN Sales.Store AS s  
ON c.CustomerID = s.BusinessEntityID ;
```

Um exemplo da obrigatoriedade do uso de *Alias* no nome da tabela, em um Select com JOIN, é quando mais de uma coluna da consulta faz referência à mesma tabela e à mesma PK.

## Avaliação 1

Prof. M.Sc. Leandro Colevati dos Santos

Exemplo:

```

create table pais(
id int identity not null primary key,
nome varchar(100))

create table filhos(
id int identity(1001,1) not null primary key,
nome varchar(100),
pai int,
mae int
foreign key (pai) references pais (id),
foreign key (mae) references pais (id))

select filhos.nome, pai.nome as 'Nome do pai', mae.nome as 'Nome da mãe'
from pais as pai
inner join filhos
on pai.id = filhos.pai
inner join pais as mae
on mae.id = filhos.mae

```

Tabela Pais			Tabela Filhos					Consulta com <i>Alias</i>			
	id	nome		id	nome	pai	mae		nome	Nome do pai	Nome da mãe
1	1	Pai 1	1	1001	Filho 1	1	2	1	Filho 1	Pai 1	Mae 1
2	2	Mae 1	2	1002	Filho 2	3	4	2	Filho 2	Pai 2	Mae 2
3	3	Pai 2	3	1003	Filho 3	5	6	3	Filho 3	Pai 3	Mae 3
4	4	Mae 2	4	1004	Filho 4	7	8	4	Filho 4	Pai 4	Mae 4
5	5	Pai 3	5	1005	Filho 5	9	10	5	Filho 5	Pai 5	Mae 5
6	6	Mae 3	6	1006	Filho 6	11	12	6	Filho 6	Pai 6	Mae 6
7	7	Pai 4	7	1007	Filho 7	13	14	7	Filho 7	Pai 7	Mae 7
8	8	Mae 4	8	1008	Filho 8	15	16	8	Filho 8	Pai 8	Mae 8
9	9	Pai 5	9	1009	Filho 9	17	18	9	Filho 9	Pai 9	Mae 9
10	10	Mae 5	10	1010	Filho 10	19	20	10	Filho 10	Pai 10	Mae 10
11	11	Pai 6									
12	12	Mae 6									
13	13	Pai 7									
14	14	Mae 7									
15	15	Pai 8									
16	16	Mae 8									
17	17	Pai 9									
18	18	Mae 9									
19	19	Pai 10									
20	20	Mae 10									

Verifique que demos o *alias* pai para a tabela pais e, também demos o *alias* mãe para a tabela pais. Com isso, conseguimos fazer um Inner Join com 2 colunas que referenciam a mesma tabela e a mesma PK.