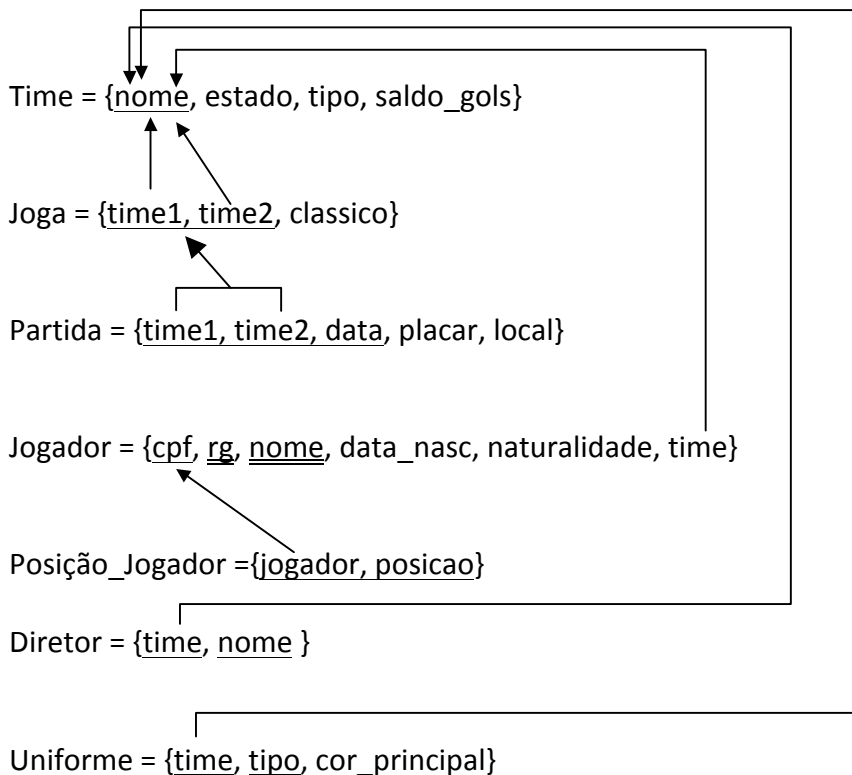


Prática 2

Exercícios SQL - DML (SGBD Oracle 19c)

Considere o esquema abaixo (o mesmo da Prática 1):



Nesse esquema, as seguintes restrições semânticas são conhecidas:

- o **tipo** do **Time** só pode assumir os valores 'AMADOR' e 'PROFISSIONAL';
- o atributo **classico** de **Joga** indica se o jogo é um clássico ou não;
- um jogador deve obrigatoriamente atuar em um time;
- o **tipo** do Uniforme pode ser 'TITULAR' ou 'RESERVA';
- o placar inicial (*default*) de uma partida é sempre 0X0;

LEIA TODAS AS INSTRUÇÕES ANTES DE RESOLVER OS EXERCÍCIOS!

OBS: todos os comandos para teste estão no arquivo **pratica2.sql** no Repositório do Tidia.

SET UP INICIAL...

- 1) **Dicionário de Dados:** no Oracle, todos os dados do dicionário são armazenados em tabelas e “disponibilizados” para os usuários como visões (*views*). A documentação do Oracle traz uma lista das visões (e respectivas estruturas) que compõem o dicionário de dados (<https://docs.oracle.com/en/database/oracle/oracle-database/19/refrn/index.html> - *Database Reference (Part II - Static Data Dictionary Views)*).

Teste: `select * from user_tables;`

2) Preparando a base de dados....

- remova todas as tabelas criadas na Prática 1
 - **Dica:** usando o comando abaixo, crie um *script* de remoção para todas as suas tabelas (o *script* será o resultado da consulta – rode com o **Executar Script** que fica mais fácil copiar o resultado).


```
select 'drop table '||table_name||' cascade constraints;' from user_tables;
```

(|| é um operador de concatenação)

- crie novas tabelas executando o *script* **Esquema_Futebol.sql** (disponível no Tidia)
 - Menu: *File/Open* (ou abra como arquivo texto e faça *copy/paste*)
 - Selecione a conexão apropriada no lado direito da barra de ferramentas (se necessário)
 - Execute todo o *script* com o botão de **Executar Script**
- veja com atenção a estrutura e as restrições das tabelas criadas, e compare com o que você implementou na Prática 1 (principalmente FK, UNIQUE e CHECK). Veja também inserção de **NULL** e **DEFAULT** em Partida.
- **alimente a base** com o *script* **Dados_Futebol.sql** (disponível no Tidia)

IMPORTANTE: quando uma **sessão de conexão** é iniciada, o SGBD inicia, implicitamente, uma **transação**. Os comandos SQL executados são operações dessa transação, e só são efetivados (resultados tornam-se persistentes) quando a transação termina. A transação termina com sucesso quando:

- 1) um comando DDL é executado (auto-commit);
- 2) quando o usuário desconecta da base com sucesso (transação termina implicitamente);
- 3) quando o comando **commit** é executado explicitamente.

 Assim, para que as inserções, atualizações e remoções sejam efetivadas no banco, é necessário finalizar a transação com o comando **commit**.

TESTE SOBRE OTIMIZADOR DE CONSULTAS E USO DE ÍNDICES...

- a) Conforme mencionado em aula, um índice é criado automaticamente para cada tabela. Considerando a tabela de **Time**: Qual a coluna usada como chave de indexação? É um índice de chave única ou de chave com repetição? Qual o nome do índice? Essas informações podem ser encontradas via interface do SQL Developer (clique duplo sobre a tabela) ou via consulta ao Dicionário de Dados.
- b) Como o seu esquema acabou de ser criado, ainda **não há estatísticas coletadas para ele**. Portanto, qualquer plano de consulta será gerado sem o uso de estatísticas, considerando apenas regras do otimizador e avaliações de custo. As estatísticas podem ser coletadas por meio do comando abaixo, que executa um procedimento do Oracle para **geração de estatísticas** a respeito das tabelas do esquema. Se o comando não for executado explicitamente, **as estatísticas são geradas automaticamente durante a noite** (quando o Oracle está configurado para isso).

```
EXEC DBMS_STATS.GATHER_SCHEMA_STATS(NULL, NULL);
```

- c) Execute, um a um, os comandos abaixo e observe os planos gerados, mais especificamente **o uso ou não do índice em consultas envolvendo o atributo Nome**. **Por que em alguns casos o índice é usado e em outro não?** Explique...

```
EXPLAIN PLAN FOR
```

```
select * from time where nome = 'INTER';
```

```
SELECT plan_table_output  
FROM TABLE(dbms_xplan.display());
```

```
EXPLAIN PLAN FOR
```

```
select * from time where upper(nome) = 'INTER';
```

```
SELECT plan_table_output  
FROM TABLE(dbms_xplan.display());
```

```
EXPLAIN PLAN FOR
```

```
select * from time where nome like 'IN%';
```

```
SELECT plan_table_output  
FROM TABLE(dbms_xplan.display());
```

EXPLAIN PLAN FOR

```
select * from time where nome like '%TER';
```

```
SELECT plan_table_output  
FROM TABLE(dbms_xplan.display());
```

EXERCÍCIOS...

- 1) Elabore comandos SQL para as consultas abaixo. Insira mais dados nas tabelas para verificar se os resultados das consultas estão corretos. Elabore casos de teste para testar se realmente as consultas funcionam para casos gerais e casos especiais.

SOBRE OS DADOS: assumo que nas tabelas **Joga** e **Partida** um mesmo par de times não será inserido 2 vezes com ordem trocada (por exemplo, não será permitida inserção de <Santos, Palmeiras> e <Palmeiras, Santos>). Esta restrição poderia ser implementada, por exemplo, via *procedure* ou *trigger* (não é necessário implementar nessa prática).

ÚTIL: é possível fazer junção de uma tabela com ela mesma (ou usar uma mesma tabela mais de uma vez na cláusula **FROM**), repetindo a tabela com "alias" diferentes.

Ex: `select ... from Tabela T1, Tabela T2 ...`

Q1: Selecione as partidas realizadas em Santos.

- **OBS:** observe que **não são** pedidas todas as informações das partidas e, portanto, a consulta deve retornar **apenas o que for necessário para diferenciar uma partida da outra**.

Q2: Selecione data e local de todas as partidas jogadas pelo Palmeiras;

- teste o retorno do atributo de data sem formatação nenhuma (selecionando o atributo direto) e com formatação (Pesquise a função `to_char`).
- **OBS:** verifique no *script* de inserção (Dados_Futebol.sql) o formato utilizado para inserção das datas. **Por que a consulta sem formatação retorna os dados em formato diferente e aparentemente incompletos?**

Q3: Para cada jogador, selecione cpf, nome, idade, e nome e estado do time para o qual joga. Elabore a consulta “passo-a-passo” e avalie a resposta de cada passo, da seguinte maneira:

- 1) Faça um `select *` e uma junção das tabelas necessárias **sem a condição de junção** (ou seja, um produto cartesiano). **OBS:** não é possível usar operador JOIN nesse exercício.
- 2) Inclua a condição de junção adequada.
- 3) Substitua o `*` pela lista de atributos pedida na consulta

Q4: Para todos os jogadores que jogam em times do estado de SP, selecione cpf, nome, data de nascimento e nome do time. Faça duas versões dessa consulta:

1. Usando somente cláusulas **FROM** e **WHERE** para junção
2. Usando operador **JOIN** para junção

Q6: Selecionar, para todas as partidas realizadas em Santos, os nomes dos times, data da partida, placar e se é um clássico ou não.

Q7: Selecionar, para todos os confrontos clássicos (em Joga), nome e estado de cada um dos times.

Q8: Selecionar cor do uniforme titular de todos os times profissionais do estado de MG.

OBS: apensar do identificador do time não ser pedido explicitamente, a resposta só será interpretável se retornar o identificador do time associado à cor de uniforme

Q9: Selecionar as partidas em que pelo menos um time do estado de SP jogou.

Q10: Selecionar times que nunca jogaram em São Carlos e em Belo Horizonte.

OBS: essa consulta é similar à **consulta 11 da Aula 15**.

- Teste inicialmente a resolução incorreta (erro na lógica) apresentada na lousa durante a aula e tente entender o porquê do resultado errado.
- Implemente a consulta corretamente.