

Trabalho 4

Relatório

SCC-541 Laboratório de Bases de Dados

Leonardo Gonçalves Chahud - 5266649
Murilo Franchi - 9790760
Rafael Dantas - 12563686

Prof. Dr Caetano Traina Jr.
PAE: Igor Alberte R. Eleutério

Exercício 1

Obtenha uma relação que contenha, para cada volta de corrida:

- o nome e o ano da corrida;
- o número da volta (atributo lap de LAPTICES);
- o tempo mínimo e máximo daquela volta;
- o nome e sobrenome de cada piloto que atingiu o tempo mínimo ou máximo naquela volta;
- o tempo gasto em milissegundos, naquela volta.

Dica: use a tabela LAPTICES.

```
select distinct(r.name, r.year, l.lap) as race_year_lap,  
       first_value(d.forename || ' ' || d.surname) over(partition by  
r.name, r.year, l.lap order by l.milliseconds) as name,  
       min(l.milliseconds) over(partition by r.name, r.year, l.lap order  
by r.year) as min_time,  
       first_value(d.forename || ' ' || d.surname) over(partition by  
r.name, r.year, l.lap order by l.milliseconds desc) as name,  
       max(l.milliseconds) over(partition by r.name, r.year, l.lap order  
by r.year) as max_time  

```

Exercício 2

1. Faça uma contagem do número de vitórias por Escuderia (Construtor), apresentando o nome e nacionalidade dela. Nesta parte, não é necessário utilizar funções Window Functions.

```
select distinct(c.name), c.nationality,  
       count(*) over(partition by c.name) as constructor_wins  
from results r  
join constructors c on c.constructorid = r.constructorid and  
r.position = 1  
order by constructor_wins desc;
```

2. Depois, apresente em cada tupla o número total de vitórias da nacionalidade em ordem decrescente. (Utilize Window Functions).

```
select distinct(c.name), c.nationality,  
       count(*) over(partition by c.name) as constructor_wins,  
       count(*) over(partition by c.nationality) as nationality_wins  
from results r  
join constructors c on c.constructorid = r.constructorid and  
r.position = 1  
order by constructor_wins desc;
```

3. A seguir, apresente uma coluna com o ranking de cada escuderia dentro do seu país, de forma que a ordem seja estabelecida pelo número de vitórias e o desempate seja pela ordem alfabética do nome da Escuderia. Apresente o resultado ordenado pelas vitórias da nacionalidade em ordem decrescente e, depois, pela ordem da Escuderia de forma crescente. (Utilize Window Functions).

```
select *, dense_rank() over(partition by k.nationality order by  
k.constructor_wins desc,k.name) as national_ranking  
from(  
    select distinct(c.name), c.nationality,  
           count(*) over(partition by c.name) as constructor_wins,  
           count(*) over(partition by c.nationality) as nationality_wins  
    from results r  
    join constructors c on c.constructorid = r.constructorid and  
r.position = 1  
    order by nationality_wins desc) k  
order by k.nationality, national_ranking
```

Exercício 3

Para cada corrida e piloto, apresente o tempo médio dos pit stops, ranqueando-os por sua duração em ordem crescente. Apresente:

- o nome e o ano das corridas,
- o nome completo dos pilotos,
- o tempo médio do piloto em pit stops naquela corrida,
- e seu rank na corrida.

```
select distinct(r.name) as race_name, r.year, d.forename || ' ' ||  
d.surname as name, re.position,  
    avg(p.milliseconds) over(partition by r.raceid, re.raceid,  
p.raceid, d.driverid, re.driverid, p.driverid) as average_pit_stops  
from races r  
join pitstops p on p.raceid = r.raceid  
join driver d on d.driverid = p.driverid  
join results re on re.driverid = p.driverid and re.raceid =  
p.raceid  
order by average_pit_stops;
```

Exercício 4

Liste, para cada país que tenha ao menos uma Escuderia, quais são as suas Escuderias. Para esse exercício, as Escuderias devem ser apresentadas como um único atributo cujo valor é um array com os nomes das escuderias. Dica: use a Window Function Array Agg.

```
select distinct(c.nationality),  
       array_agg(c.name) over(partition by c.nationality  
                             from constructors c  
                             order by c.nationality);
```

Exercício 5

Para cada corrida, apresente, para cada piloto:

- o identificador Id, nome e ano da corrida;
- o total de tempo da corrida;
- nome e sobrenome do piloto;
- tempo do piloto na corrida (em milissegundos);
- a diferença entre seu tempo e a do piloto que ganhou a corrida (em milissegundos);
- a desvantagem entre seu tempo e a do piloto que terminou a corrida imediatamente antes dele (em milissegundos).

O ganhador de cada corrida tem diferença nula do anterior e de quem estiver à sua frente.

Dica: Avalie o uso da função LAG.

```
select distinct(r.raceid) as race_id, r.name as race_name, r.year,
       max(re.milliseconds) over(partition by r.raceid, re.raceid) as
race_total_time,
       d.forename || ' ' || d.surname as driver_name, re.milliseconds,
       re.milliseconds - lag(re.milliseconds, 1) over(partition by
r.raceid, re.raceid order by re.milliseconds) as
disadvantage_next_driver,
       re.milliseconds - min(re.milliseconds) over(partition by r.raceid,
re.raceid) as disadvantage_first
  from races r
  join results re on re.raceid = r.raceid and re.statusid = 1
--status 1 == finished
  join driver d on d.driverid = re.driverid;
```