

# SQL

## Vistas e Combinação de Queries



INFORMÁTICA  
1º ANO – 2023/2024

ARTUR JORGE DA SILVA ROCHA  
E-MAIL: D001457@UMAIA.PT

# Vista

2

- Uma vista é uma tabela virtual criada a partir de um comando SELECT
- Sistema guarda apenas definição da vista. Dados são calculados em tempo de execução

# Vantagens das Vistas

3

- Simplificação de uma query
- Independência lógica dos dados
- Segurança adicional
- Separação/personalização dos dados vistos por cada utilizador
- Simplificação da estrutura da base de dados

# Desvantagens das Vistas

4

- Pior eficiência
- Alterações à base de dados podem invalidar vistas definidas previamente
- Não há transparência completa: muitas vistas não aceitam comandos de actualização dos dados

# Criação de Vistas

5

- As vistas podem ser criadas com o comando **CREATE VIEW**
- Sintaxe: **CREATE VIEW <nome> [coluna1, [coluna2, ...]] AS SELECT...**

# Exemplo

6

- Resposta à pergunta: em que anos teve a McLaren mais pontos que a Ferrari?
- Vista que devolve os pontos de cada equipa num dado ano

CREATE VIEW PontosEquipas AS

SELECT epoca.ano, construtor.nome + '-' + motor.nome equipa,  
sum(pontos) pontos

FROM construtor JOIN equipa

ON construtor.id = equipa.construtor\_id

JOIN epoca

ON equipa.ano = epoca.ano

AND epoca.piloto\_id IN(equipa.piloto1\_id, equipa.piloto2\_id)

JOIN motor

ON equipa.motor\_id = motor.id

GROUP BY epoca.ano, construtor.id, construtor.nome, motor.nome

# Exemplo (cont.)

7

- **Query principal**

```
SELECT ml.ano, ml.equipa McLaren, ml.pontos,  
fe.equipa Ferrari, fe.pontos  
FROM PontosEquipas ml, PontosEquipas fe  
WHERE ml.ano = fe.ano  
AND ml.equipa like 'McLaren%'  
AND fe.equipa like 'Ferrari%'  
AND ml.pontos > fe.pontos
```

# Exemplo (cont.)

8

- Solução alternativa (sem vistas)

```
SELECT ep1.ano, c1.nome + '-' + m1.nome McLaren, SUM(ep1.pontos)
pontos,
      c2.nome + '-' + m2.nome Ferrari, SUM(ep2.pontos) pontos
FROM equipa e1 join epoca ep1 on e1.ano = ep1.ano
      join construtor c1 on e1.construtor_id = c1.id
      join motor m1 on e1.motor_id = m1.id,
      equipa e2 join epoca ep2 on e2.ano = ep2.ano
      join construtor c2 on e2.construtor_id = c2.id
      join motor m2 on e2.motor_id = m2.id
WHERE
      (( e1.piloto1_id = ep1.piloto_id AND e2.piloto1_id = ep2.piloto_id )
      OR ( e1.piloto2_id = ep1.piloto_id AND e2.piloto2_id = ep2.piloto_id ))
AND c1.nome like 'McLaren'
AND c2.nome like 'Ferrari'
AND e1.ano = e2.ano
GROUP BY ep1.ano, c1.nome, c2.nome, m1.nome, m2.nome
HAVING SUM(ep1.pontos) > SUM(ep2.pontos)
ORDER BY ano
```



# Operações de Conjuntos

9

# Queries e conjuntos

10

- Uma tabela é um conjunto de registos de um dado domínio
- Uma query é um multi-conjunto de registos de um dado domínio
- Por isso, os resultados de várias queries podem ser combinados com operadores de conjuntos
- Operadores: UNION, INTERSECT, EXCEPT
- Ambas as queries têm de devolver registos com o mesmo tipo de dados na mesma ordem

# UNION

11

- Devolve a união dos registos de duas queries
- Não devolve linhas repetidas (união de conjuntos elimina repetidos)
- Se resultado for ordenado, ordenação afecta ambas as queries
- Tuplos podem vir de tabelas diferentes e ter significados diferentes

# UNION

12

- Exemplo:
  - Seleccionar todos os construtores e pilotos, e a respectiva nacionalidade, ordenando por país  
(SELECT pname + ' ' + unome, pais  
FROM piloto)  
**UNION**  
(SELECT nome, pais  
FROM construtor)  
ORDER BY pais

# INTERSECT

13

- Devolve apenas os registos que aparecem em ambas as queries
- Exemplo:
  - Mostrar os países que têm pilotos e construtores  
(SELECT pais  
FROM piloto)  
**INTERSECT**  
(SELECT pais  
FROM construtor)  
ORDER BY pais

# EXCEPT

14

- Devolve apenas os registos que aparecem na primeira query, mas não na segunda
- Exemplo:
  - Mostrar todos os países onde já tenham nascido pilotos mas que não tenham construtores

SELECT pais

FROM piloto

**EXCEPT**

SELECT pais

FROM construtor

ORDER BY pais

# Exercícios

15

# Exercícios (I)

16

- Mostrar uma lista dos países que para cada ano tiveram equipas E pilotos da sua nacionalidade a correr
- Listar o nome dos pilotos que **só** participaram em campeonatos quando tinham entre 25 e 30 anos, inclusive. Ou seja, não mostrar os pilotos que participaram em campeonatos quando tinham mais de 30 ou menos de 25 anos.
- Mostrar uma tabela listando, para cada país e cada ano, os construtores e os pilotos desse país que participaram no campeonato desse ano. Ordenar primeiro por ano, depois por país, depois por tipo de participante (Piloto ou Construtor). Eliminar registos onde o país não esteja definido.
- Criar uma vista que devolva para cada ano todas as equipas, mas sem nenhuma das chaves alheias da tabela equipa. Em vez disso, deve mostrar o nome completo de cada piloto, o nome completo de uma equipa (construtor e motor) e ainda os dados respectivos de cada piloto e da equipa: nacionalidades, estatísticas de cada piloto em cada ano, anos de nascimento e fundação.



# Exercícios (II)

17

- Criar uma vista que liste o total de grandes prêmios ganhos por pilotos de cada nacionalidade em cada ano
- Criar uma vista que para cada ano indique o piloto que conseguiu mais pontos
- Criar uma vista que para cada ano indique o construtor que conseguiu mais pontos
- Dar uma lista de CAMPEONATOS ganhos por pilotos de cada nacionalidade
- Mostrar os anos em que uma equipa conseguiu o título de construtores e de pilotos
- Mostrar para cada ano o vencedor do título de construtores e o vencedor do título de pilotos, indicando o nome, a nacionalidade e a idade (do piloto e do construtor).
- Mostrar as equipas que já ganharam campeonatos de construtores que nunca tenham ganho um campeonato de pilotos