

# SQL

## SELECT – JOINs



**INFORMÁTICA**  
**1º ANO – 2023/2024**

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

# Join (Junção)



- Em bases de dados normalizadas, a maioria das queries precisa de extrair dados de mais do que uma tabela
- Não basta pôr mais tabelas no FROM → resultado seria um produto cartesiano
- É necessário efectuar a junção das diversas tabelas: esta operação designa-se por Join ou Junção

# Produto Cartesiano



- Devolve todas as combinações possíveis entre os registos de todas as tabelas, mostrando uma linha com os dados todos de um registo da primeira tabela, depois todos os atributos de outro registo da segunda tabela e assim sucessivamente
- Inclui combinações automaticamente contraditórias, por incluírem valores diferentes para colunas que representam a mesma informação.
- Exemplo: `SELECT * FROM continente, pais`

# Join vs Produto Cartesiano



- O Join elimina do produto cartesiano as linhas com informação contraditória
- Isto consegue-se unificando os valores de duas colunas em tabelas diferentes que representam a mesma informação
- Esta unificação garante que em ambas as colunas o valor é o mesmo, e pode ser feita no WHERE ou num JOIN
- Cada unificação destas junta duas tabelas. Se a query inclui  $n$  tabelas, tem de haver  $n - 1$  junções

# Join com WHERE



- Para especificar o Join através da cláusula WHERE, basta indicar os campos que têm de ser iguais.

```
SELECT * FROM piloto, pais
```

```
WHERE piloto.pais = pais.sigla
```

- SELECT com mais do que uma tabela: imaginar um cursor a percorrer os registos de cada tabela. Join indica quais as posições legais para cada cursor.

# Self-Join



- Junção de uma tabela consigo mesma (auto-junção).
- Exemplo:
  - mostrar todos os pares de pilotos que têm o mesmo último nome

```
SELECT p1.pnome + ' ' + p1.unome, p2.pnome + ' ' +  
p2.unome  
FROM piloto p1, piloto p2  
WHERE p1.unome = p2.unome  
AND p1.id < p2.id
```

# Cópias da mesma tabela num SELECT



- Comparação de dois registos da mesma tabela entre si (slide anterior)
- Necessidade de referência a registos diferentes da mesma tabela, mas sem serem comparados directamente: ambos devem ser referidos por registos provindos de outras tabelas

# Cópias da mesma tabela num SELECT



- Exemplo:
  - para cada ano, mostrar o nome dos pilotos que correram para uma dada equipa

```
SELECT ano, construtor.nome, p1.pnome + ' ' +  
p1.unome, p2.pnome + ' ' + p2.unome  
FROM construtor, equipa, piloto AS p1, piloto AS p2  
WHERE equipa.construtor_id = construtor.id  
AND equipa.piloto1_id = p1.id  
AND equipa.piloto2_id = p2.id  
ORDER BY ano, nome
```



# Cópias da mesma tabela num SELECT



- Exemplo

- mostrar o nome, a idade e o nome do respectivo país de todos os construtores
- ordenar por ordem alfabética do nome de país e, para cada país, por ordem alfabética

```
SELECT c.nome, year(getdate()) - c.anoFund as idade,  
       p.nome
```

```
FROM construtor c, pais p
```

```
WHERE c.pais = p.sigla
```

```
ORDER BY p.nome, c.nome
```

# A cláusula JOIN



- JOIN é a forma correcta de fazer uma junção de acordo com a norma SQL
- É mais flexível, permitindo opções do tipo INNER JOIN e OUTER JOIN
- Separa as condições de junção das condições de filtragem de registos, tornando a query mais compreensível
- Exemplos anteriores com WHERE serviram apenas para explicar o conceito da junção

# Exemplo de junção com JOIN



```
SELECT e.ano, c.nome + ' - ' + motor.nome, c.pais  
FROM equipa e  
      JOIN construtor c ON e.construtor_id = c.id  
      JOIN motor ON e.motor_id = motor.id  
WHERE ano > 1960  
ORDER BY ano
```

# INNER JOIN



- Também escrito apenas JOIN. É, portanto, o comportamento por omissão
- Indica a tabela a juntar e a condição de junção, na cláusula ON
- Condição de junção pode usar qualquer operador de comparação de campos
- Operador mais usual é a igualdade
- Como funciona?
  - FROM a JOIN b ON a.valA = b.valB:
  - Resultado inclui apenas as linhas de a para as quais existe uma linha de b que tenha em valB o mesmo valor de valA.
  - Valores de a ou b que não tenham o correspondente na outra tabela simplesmente não aparecem

# OUTER JOIN



- Existe em três versões: LEFT OUTER JOIN, RIGHT OUTER JOIN e FULL OUTER JOIN
- Resultado inclui valores de uma tabela que não tenham correspondente na outra tabela
- Como funciona?
  - FROM a LEFT OUTER JOIN b ON a.valA = b.valB:
  - Resultado inclui todas as linhas incluídas por INNER JOIN
  - Inclui também uma linha para todos os outros registos de a.
  - Se não existir um valor em b pelo qual se possa fazer a junção, os atributos de b aparecem todos iguais a *null*.

# OUTER JOIN



- Para incluir todos os registos da segunda tabela (neste caso b) usa-se RIGHT OUTER JOIN
- Pode-se escrever FULL OUTER JOIN para incluir todos os registos de ambas as tabelas
- A palavra OUTER pode ser omitida em qualquer dos casos.

# OUTER JOIN - Exercício



- Executar a seguinte query para perceber as diferenças entre os vários tipos de JOIN:
  - INNER JOIN devolve 12 linhas
  - LEFT JOIN devolve 14 linhas
  - RIGHT JOIN devolve 34 linhas
  - FULL JOIN devolve 36 linhas

**SELECT \* FROM construtor**

**JOIN pais ON construtor.pais = pais.sigla**

**ORDER BY id**

# Exercícios



- Mostrar todos os países diferentes de onde sejam nacionais pilotos nascidos depois de 1960.
- Obter uma lista de todos os motores diferentes usados depois de 1970
- Mostrar todos os construtores fundados num país cujo nome se escreva com pelo menos 6 letras. Indique também o nome do país
- Mostrar os nomes de todas as duplas de pilotos de uma mesma nacionalidade que, num dado ano, estavam na mesma equipa, juntamente com o nome da equipa (construtor-motor) e do respectivo país.
- Mostrar todos os pares de pilotos que tenham nascido no mesmo ano.
- Mostrar todos os pilotos alemães que já correram em carros com motor BMW, indicando o nome do construtor e em que ano o fizeram.



# Exercícios



- Mostrar todos os pilotos que correram para marcas da sua nacionalidade, sem repetições.
- Mostrar os pilotos, e respectiva nacionalidade, que correram em equipas fundadas, na altura, havia mais de 20 anos. Indicar o nome da equipa, o ano de fundação e o ano em que correram.
- Mostrar o nome das equipas que num dado ano tiveram dois pilotos de nacionalidades diferentes da sua.
- Mostrar o nome e a nacionalidade de pilotos que tenham corrido em carros não ingleses até 2000, ordenando por idade dos pilotos, do mais velho para o mais novo.
- Mostrar as equipas que tenham tido num dado ano (e indicar o ano) dois pilotos com pelo menos 30 anos. Indicar o último nome dos pilotos e a respectiva idade.

# Exercícios



- Mostrar para cada ano o total de pontos de cada equipa, ordenando as equipas por ano e por pontuação
- Mostrar para cada piloto e por cada ano a média de pontos por corrida começada.
- Mostrar os anos, os pilotos e a equipa em que o segundo piloto teve mais vitórias que o primeiro (da mesma equipa)
- Mostrar por cada ano uma lista de pilotos, indicando a sua nacionalidade e nome completo da equipa, ordenada pelo número de pontos que cada um conseguiu.
- Mostrar o nome no formato “<inicial>. <último nome>” de todos os pilotos não europeus