Universidade do Minho

Ano Letivo: 2023/24
Turno: PL9

Bases de Dados

PL09 – Exploração Simples e Avançada

Docente: Cristiana Neto
Email: cristiana.neto@algoritmi.uminho.pt

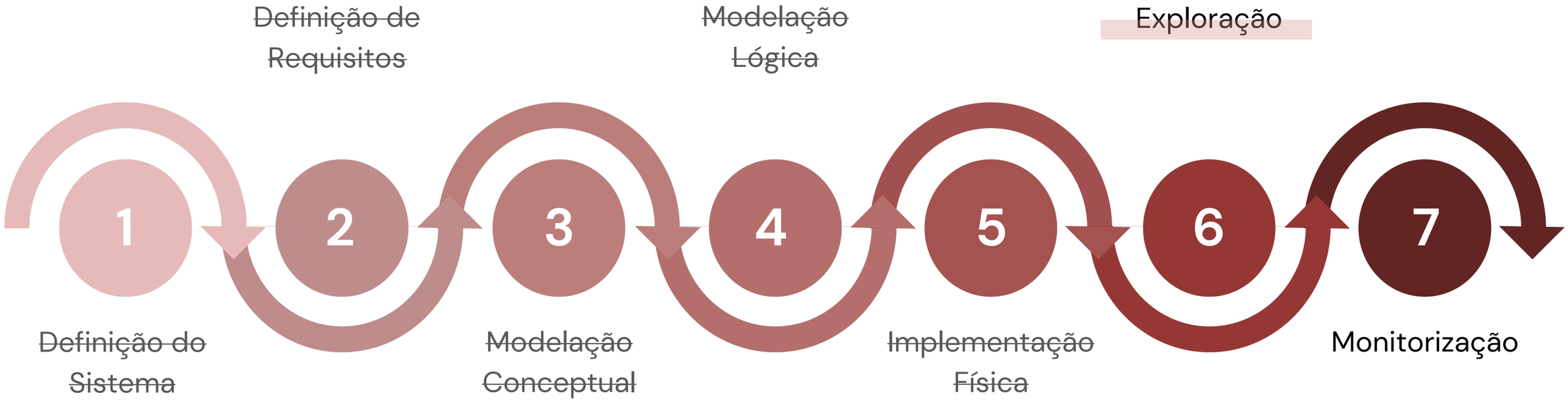
Horário de Atendimento:
6ª feira 09h-10h

Sumário

Bibliografia:

- Connolly, T., Begg, C., Database Systems, A Practical Approach to Design, Implementation, and Management , Addison-Wesley, 4a Edição, 2004. **(Chapter 6 e 7)**
- Belo, O., "Bases de Dados Relacionais: Implementação com MySQL", FCA – Editora de Informática, 376p, Set 2021. ISBN: 978-972-722-921-5. **(Capítulo 4 e 5)**

Ciclo de vida de um SBD



FASE 6: Exploração

Expressões Regulares

As expressões regulares diferenciam-se do operador LIKE por permitirem construir padrões mais flexíveis. No entanto, o tempo de consulta pode aumentar caso se usem padrões complexos. Estas expressões usam os operadores **RLIKE** ou **REGEXP**. Alguns metacaracteres comumente usados numa expressão regular:

^	Corresponde à posição no início da string pesquisada;
$	Corresponde à posição no final da string pesquisada;
.	Corresponde a qualquer caracter;
[...]	Corresponde a qualquer caracter específicado dentro dos parêntesis rectos;
[^...]	corresponde a qualquer caractere não especificado dentro dos parêntesis
p1|p2	corresponde a qualquer um dos padrões p1 ou p2
{n}	corresponde a n número de instâncias do caractere anterior
{m,n}	corresponde de m a n número de instâncias do caractere anterior

FASE 6: Exploração

➡ Expressões Regulares

EXEMPLOS:

- **Retorna os atletas cujo nome comece com a letra ‘i’ ou ‘a’:**

SELECT nome FROM atletas WHERE nome REGEXP ‘^[ia]';

SELECT nome FROM atletas WHERE nome REGEXP ‘^i|^a';

- **retorna as especialidades cujo nome não termina com as letras ‘gia’**

SELECT descricao FROM especialidades WHERE descricao NOT REGEXP 'gia$';

- **retorna todos as especialidades que contêm os caracteres ‘is’**

SELECT descricao FROM especialidades WHERE descricao REGEXP 'is';

FASE 6: Exploração

➡ Expressões Regulares

- retorna as especialidades que contêm extamente 10 caracteres:

SELECT descricao FROM especialidades WHERE descricao REGEXP '^.{10}$';

SELECT descricao FROM especialidades WHERE descricao REGEXP '^.........$';

- retorna as especialidades que contêm entre 5 a 10 caracteres:

SELECT descricao FROM especialidades WHERE descricao REGEXP '^.{5,10}$';

- retorna os atletas que contêm uma letra entre 'a' e 'c', seguidas por qualquer caracter, seguidas pela letra 'a'.

SELECT nome FROM atletas WHERE nome REGEXP '[a-c].[a]';

- retorna todas as modalidades que comecem com vogal ou terminem em 'ol'

SELECT descrição FROM modalidades WHERE descrição REGEXP '^[aeiou]|ol$';

FASE 6: Exploração

➡ SUBQUERIES

Uma instrução SELECT pode ser usada dentro de outra instrução SELECT, é chamada de SELECT interna (ou subselect) e tem de estar entre parêntesis curvos. Por sua vez, um subselect pode ser usado dentro de outro subselect.

A) subselect com operadores de comparação

Qual são os profissionais com mais artigos científicos?

```
SELECT * FROM Profissionais_saude WHERE num_artigos_publicados = (SELECT MAX(num_artigos_publicados) FROM Profissionais_saude);
```

B) subselect com operadores IN e NOT IN

Listar os treinadores que ainda não realizaram nenhum treino.

```
SELECT * FROM Treinadores WHERE idTreinador NOT IN (SELECT idTreinador FROM Treinos)
```

FASE 6: Exploração

➡ SUBQUERIES

C) subselect na cláusula FROM – o conjunto de resultados retornado de um subselect é usado como uma tabela temporária.

Listar o número máximo e o número mínimo de treinos realizados por um só atleta.

SELECT max(total),min(total) from (select idAtleta, COUNT(*) as total FROM Treinos GROUP BY idAtleta) as sub;

D) subselect com operadores EXISTS e NOT EXISTS

Listar os treinadores que realizaram treinos.

SELECT * FROM treinadores t WHERE EXISTS (SELECT * FROM treinos c WHERE c.idTreinador = t.idTreinador);

FASE 6: Exploração

SUBQUERIES

E) subselect com operadores ANY e ALL: É usado para efetuar uma comparação entre o valor de uma coluna e uma range de valores. Ambos retornam um valor booleano como resultado. O ANY retorna TRUE se QUALQUER um dos valores da subconsulta atender à condição. O ALL retorna TRUE se TODOS os valores da subconsulta atenderem à condição .

Listar os treinadores que realizaram treinos.

```
SELECT * FROM treinadores WHERE idTreinador =  ANY(SELECT idTreinador FROM treinos);
```

FASE 6: Exploração

➡ SUBQUERIES

```
SELECT * FROM R
WHERE r1 IN
(SELECT r1 FROM S);
```

```
SELECT * FROM R
WHERE EXISTS
(SELECT * FROM S
WHERE R.r1 = S.r1);
```

```
SELECT * FROM R
WHERE r1 = ANY
(SELECT r1 FROM S );
```

```
SELECT * FROM R
INNER JOIN S USING
(r1);
```

```
SELECT * FROM R
WHERE r1 NOT IN
(SELECT r1 FROM S);
```

```
SELECT * FROM R
WHERE NOT EXISTS
(SELECT * FROM S
WHERE R.r1 = S.r1);
```

```
SELECT * FROM R
WHERE r1 <> ALL
(SELECT r1 FROM S );
```

```
SELECT * FROM R
LEFT JOIN S USING
(r1) WHERE S.id is
NULL;
```

FASE 6: Exploração

Operações de Junção

A operação de Junção é utilizada para combinação dos dados contidos numa ou mais tabelas através das colunas em comum, ou seja, as *foreign keys*. A cláusula JOIN é usada na instrução SELECT e aparece sempre depois da cláusula FROM.

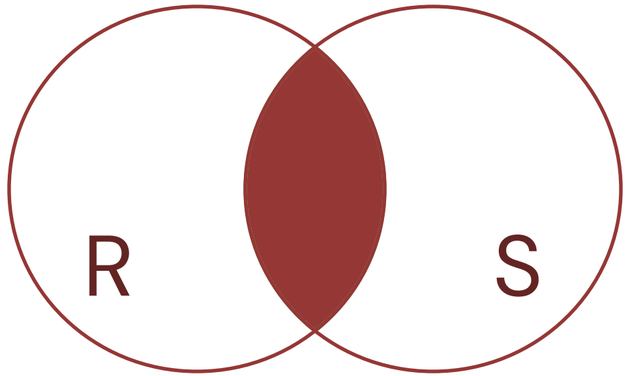
O mysql suporta diferentes operações de junção:

- CROSS JOIN;
- NATURAL JOIN;
- INNER JOIN;
- LEFT JOIN;
- RIGHT JOIN;

FASE 6: Exploração

NATURAL JOIN

A operação de Junção Natural, é uma operação entre duas relações R e S que permite inter-relacionar essas duas relações através das colunas que sejam comuns às duas relações e que possuam valores iguais. O esquema da relação resultante contém todas as colunas de ambas as relações – excluindo-se uma das colunas de junção.



$$SELECT\ *\ FROM\ R\ NATURAL\ JOIN\ S;$$

FASE 6: Exploração

NATURAL JOIN

EXEMPLOS:

- Quais são as especialidades exercidas pelos profissionais de saúde?

```
SELECT * FROM Especialidades NATURAL JOIN Profissionais_saude;
```

- Liste as modalidades dos treinadores com PhD.

```
SELECT * FROM treinadores NATURAL JOIN Modalidades WHERE grau = 'PhD';
```

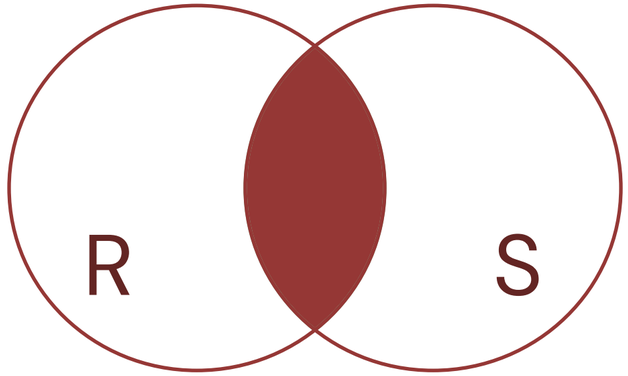
- Liste para cada agendamento, a prioridade, a descrição e o tipo de exame.

```
SELECT e.desc_exame,e.tipo,a.prioridade FROM Exames e NATURAL JOIN Agendamentos a;
```

FASE 6: Exploração

INNER JOIN

A operação de Junção Interna, é uma operação entre duas relações R e S que permite inter-relacionar essas duas relações através das colunas que satisfaçam a expressão predicativa. O esquema da relação resultante contém todas as colunas de ambas as relações.



$$SELECT * FROM\ R\ INNER\ JOIN\ S\ ON\ R.A = S.B;$$

Para além do operador de igualdade (=), podem ser usados os operadores >, < e <>.

$$SELECT * FROM\ R\ INNER\ JOIN\ S\ USING\ (A);$$

Se as colunas de junçao das duas tabelas tiverem o mesmo nome.

FASE 6: Exploração

➡ INNER JOIN

EXEMPLOS:

- Quais são as modalidades exercidas pelos treinadores?

```
SELECT * FROM Modalidades INNER JOIN Treinadores USING(cod_modalidade)
```

- Quais são os profissionais de saúde que ainda trabalham?

```
select * from Profissionais_saude p INNER JOIN Funcionários f ON f.idFuncionário = p.idProfissionalSaude
WHERE dta_fim_serv IS NULL
```

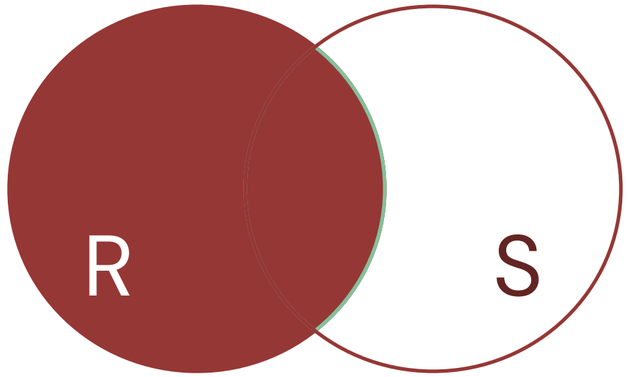
- Quais os treinos cuja descrição é igual ao nome da modalidade?

```
select * from Treinadores t
INNER JOIN Treinos c ON c.idTreinador = t.idTreinador
INNER JOIN Modalidades m ON m.cod_modalidade = t.cod_modalidade
WHERE c.descrição = m.descrição
```

FASE 6: Exploração

LEFT JOIN

A operação de Junção Externa à Esquerda (*Outer Left Join*), integra na relação final todas as tuplas da relação à esquerda, mesmo quando estas não obedecem aos critérios de junção definidos. Ou seja, os tuplos de R que não têm correspondência nas colunas comuns de S são incluídos no resultado. Quando não existem valores correspondentes na segunda relação S, apresentam-se valores nulos (NULL).

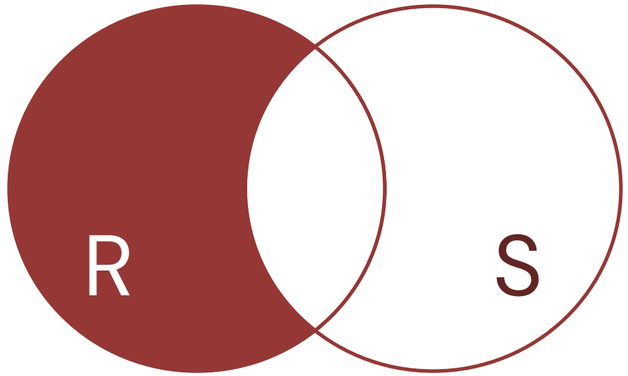


$$SELECT\ *\ FROM\ R\ LEFT\ JOIN\ S\ ON\ R.A = S.B;$$
$$SELECT\ *\ FROM\ R\ LEFT\ JOIN\ S\ USING(A);$$

FASE 6: Exploração

➡ LEFT JOIN

Como a operação de Junção Externa à Esquerda (*Outer Left Join*) integra na relação final todas as tuplas da relação à esquerda R, mesmo quando não têm correspondência na relação à direita S (ou seja apresentam-se valores nulos), é possível selecionar apenas as tuplas da relação R que não têm correspondência na relação S usando a cláusula WHERE e o operador IS NULL.

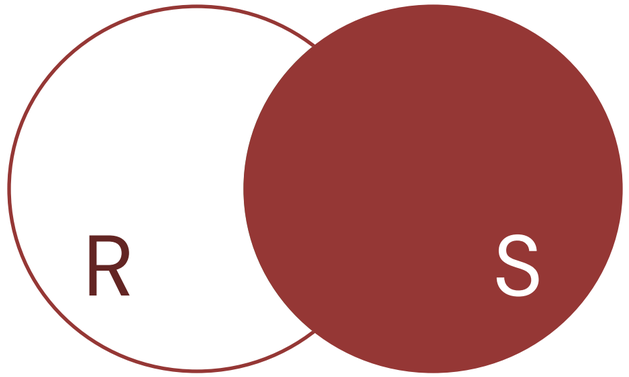



```
SELECT * FROM R LEFT JOIN S ON R.A = S.B WHERE S.ID IS NULL;
SELECT * FROM R LEFT JOIN S USING(A) WHERE S.ID IS NULL;
```

FASE 6: Exploração

RIGHT JOIN

A operação de Junção Externa à Direita (*Outer Right Join*), é semelhante Junção Externa à Esquerda, exceto que o tratamento das tabelas unidas é invertido. Ou seja, integra na relação final todas as tuplas da relação à direita, mesmo quando estas não obedecem aos critérios de junção definidos. Quando não existem valores correspondentes na primeira relação R, apresentam-se valores nulos (NULL).

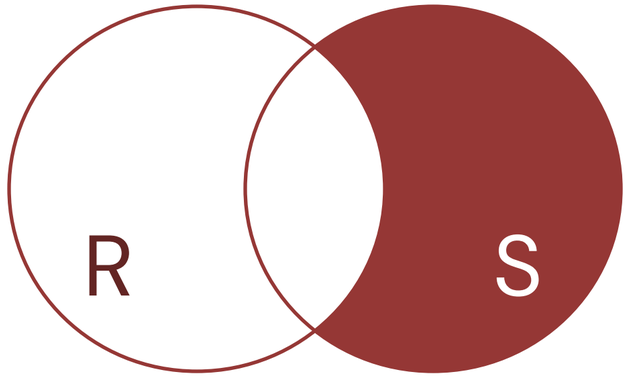


$$SELECT\ *\ FROM\ R\ RIGHT\ JOIN\ S\ ON\ R.A = S.B;$$
$$SELECT\ *\ FROM\ R\ RIGHT\ JOIN\ S\ USING(A);$$

FASE 6: Exploração

➡ RIGHT JOIN

Como a operação de Junção Externa à Direira (*Outer Right Join*) integra na relação final todas as tuplas da relação à direita S, mesmo quando não têm correspondência na relação à esquerda R (ou seja apresentam-se valores nulos), é possível selecionar apenas as tuplas da relação S que não têm correspondência na relação R usando a cláusula WHERE e o operador IS NULL.



$$SELECT * FROM\ R\ RIGHT\ JOIN\ S\ ON\ R.A = S.B\ WHERE\ R.ID\ IS\ NULL;$$
$$SELECT * FROM\ R\ RIGHT\ JOIN\ S\ USING(A)\ WHERE\ R.ID\ IS\ NULL;$$

FASE 6: Exploração

➡ LEFT/RIGHT JOIN

EXEMPLOS:

- Quais os nomes dos treinadores que nunca deram treinos?

```
select nome FROM Funcionários f
INNER JOIN Treinadores t ON t.idTreinador = f.idFuncionário
LEFT JOIN Treinos tr ON t.idTreinador = tr.idTreinador
WHERE nr_treino IS NULL

SELECT nome from Treinos tr
RIGHT JOIN Treinadores T USING(idTreinador)
INNER JOIN  Funcionários f  ON t.idTreinador = f.idFuncionário
WHERE nr_treino IS NULL
```

FASE 6: Exploração

➡ LEFT/RIGHT JOIN

EXEMPLOS:

- Quais as descrições dos exames que nunca foram agendados?

```
SELECT * FROM Exames e
LEFT JOIN Agendamentos USING (cod_exame)
WHERE nr_episodio IS NULL
```

```
SELECT * FROM Agendamentos
RIGHT JOIN Exames USING (cod_exame)
WHERE nr_episodio IS NULL
```

FASE 6: Exploração

Resolução de Exercícios

Ficha de Excercícios PL09

Universidade do Minho

Ano Letivo: 2023/24
Turno: PL9

Bases de Dados

PL09 – Exploração Simples e Avançada

Docente: Cristiana Neto
Email: cristiana.neto@algoritmi.uminho.pt

Horário de Atendimento:
6ª feira 09h-10h