# Bases de Dados

## SQL: STRUCTURED QUERY LANGUAGE (CONTINUAÇÃO)

# Sub-queries

Utilização de um SELECT dentro de outro. Representação entre parêntesis.

O resultado do SELECT “de dentro” é utilizado para o cálculo do SELECT “de fora”

Pode ser utilizada dentro do WHERE e do HAVING

Podem ser do tipo:

- Escalar: retornado um único valor
- Linha: retornada uma linha
- Tabela: retornada uma tabela

# Subquery com igualdade

*Listar os funcionários que trabalham na loja situada na 'R. Central 34'*

```
SELECT *

FROM Funcionario

WHERE idLoja = (
    SELECT idLoja
    FROM Loja
    WHERE rua='R. Central 34'
    );
```

| idFuncionario | pNome | uNome | cargo | genero | dtNascimento | salario | idLoja |
|---|---|---|---|---|---|---|---|
| SG14 | David | Ferreira | Supervisor | M | 1958-03-24 | 18000 | B003 |
| SG37 | Ana | Santos | Assistente | F | 1960-11-10 | 12000 | B003 |
| SG5 | Susana | Silva | Gerente | F | 1940-06-03 | 24000 | B003 |

# Subquery e função de agregação

*Listar os funcionários cujo salario é maior do que a média dos salários, e mostrar quanto maior é do que a média*

```
SELECT
idFuncionario, pNome, uNome, cargo, salario, salario-(SELECT AVG(salario) FROM Funcionario) AS diferenca
FROM Funcionario
WHERE salario > (
    SELECT AVG(salario)
    FROM Funcionario
    );
```

| idFuncionario | pNome | uNome | cargo | salario | diferenca |
|---|---|---|---|---|---|
| SG14 | David | Ferreira | Supervisor | 18000 | 1000.0000 |
| SG5 | Susana | Silva | Gerente | 24000 | 7000.0000 |
| SL21 | Joao | Alves | Gerente | 30000 | 13000.0000 |

# Consultas encadeadas IN

*Listar as propriedades geridas por funcionários que trabalhem na loja na 'R. Central 34'*

```
SELECT *
FROM Propriedade
WHERE idFuncionario IN (
  SELECT idFuncionario
  FROM Funcionario
  WHERE idLoja = (
    SELECT idLoja
    FROM Loja
    WHERE rua='R. Central 34')
  );
```

| idPropriedade | rua | cidade | codPostal | tipo | quartos | renda | idProprietario | idFuncionario | idLoja |
|---|---|---|---|---|---|---|---|---|---|
| PG16 | R. Sameiro 87 | Braga | 1122 | Apartamento | 4 | 450 | CO93 | SG14 | B003 |
| PG21 | R. Bom Jusus 32 | Braga | 1122 | Moradia | 5 | 600 | CO87 | SG37 | B003 |
| PG36 | Av. Central 32 | Braga | 1122 | Apartamento | 3 | 375 | CO93 | SG37 | B003 |

# ANY / SOME

*Listar todos os funcionários cujo salario é maior do que o salario de pelo menos um dos funcionários da Loja B003*

```
SELECT *
FROM Funcionario
WHERE salario >
SOME (
  SELECT salario
  FROM Funcionario
  WHERE idLoja='B003'
);
```

Salarios dos funcionários da loja B003

| idFuncionario | pNome | uNome | cargo | genero | dtNascimento | salario | idLoja |
|---|---|---|---|---|---|---|---|
| SG14 | David | Ferreira | Supervisor | M | 1958-03-24 | 18000 | B003 |
| SG5 | Susana | Silva | Gerente | F | 1940-06-03 | 24000 | B003 |
| SL21 | Joao | Alves | Gerente | M | 1945-10-01 | 30000 | B005 |

# ALL

*Selecionar os funcionários cujo salario é maior do que os salários de todos os funcionários da loja B003*

```
SELECT *
FROM Funcionario
WHERE salario >
ALL (
  SELECT salario
  FROM Funcionario
  WHERE idLoja='B003'
);
```

Salarios dos funcionários da loja B003

| idFuncionario | pNome | uNome | cargo | genero | dtNascimento | salario | idLoja |
|---|---|---|---|---|---|---|---|
| SL21 | Joao | Alves | Gerente | M | 1945-10-01 | 30000 | B005 |

# Queries com múltiplas tabelas - JOIN simples

Listar os ids e nomes de todos os clientes que visitaram uma propriedade, a propriedade visitada e os comentários feitos

Clientes:

| idCliente | pNome | uNome |
|---|---|---|
| CR56 | Adelina | Santos |
| CR62 | Maria | Pereira |
| CR74 | Miguel | Silva |
| CR76 | Joao | Alves |

Visitas e comentários:

| idCliente | idPropriedade | comentario |
|---|---|---|
| CR56 | PG36 | NULL |
| CR56 | PG4 | muito perto |
| CR56 | PA14 | demasiado pequeno |
| CR62 | PA14 | sem sala de jantar |
| CR76 | PG4 | demasiado longe |

SELECT c.idCliente, pNome, uNome, idPropriedade, comentário

FROM Cliente c, Visita v

WHERE c.idCliente = v.idCliente;

| idCliente | pNome | uNome | idPropriedade | comentario |
|---|---|---|---|---|
| CR56 | Adelina | Santos | PG36 | NULL |
| CR56 | Adelina | Santos | PG4 | NULL |
| CR56 | Adelina | Santos | PA14 | demasiado pequeno |
| CR62 | Maria | Pereira | PA14 | sem sala de jantar |
| CR76 | Joao | Alves | PG4 | demasiado longe |

# Ordenação de um JOIN

*Para cada loja, listar os números e nomes dos funcionários que gerem propriedades e as propriedades geridas*

```
SELECT f.idLoja, f.idFuncionario, pNome, uNome, idPropriedade
FROM Funcionario f, Propriedade p
WHERE f.idFuncionario = p.idFuncionario
ORDER BY f.idLoja, f.idFuncionario, idPropriedade;
```

| idLoja | idFuncionario | pNome | uNome | idPropriedade |
|---|---|---|---|---|
| B003 | SG14 | David | Ferreira | PG16 |
| B003 | SG37 | Ana | Santos | PG21 |
| B003 | SG37 | Ana | Santos | PG36 |
| B005 | SL41 | Julia | Borges | PL94 |
| B007 | SA9 | Maria | Marques | PA14 |

# JOIN com três tabelas

Para cada loja, listar os números e nomes dos funcionários que gerem propriedades, incluindo a cidade da loja e as propriedades que o funcionário gere

Loja:

| idLoja | cidade |
|---|---|
| B002 | Felgueiras |
| B003 | Braga |
| B004 | lousada |
| B005 | Felgueiras |
| B007 | Porto |

Funcionario:

| idFuncionario | pNome | uNome | idLoja |
|---|---|---|---|
| SA9 | Maria | Marques | B007 |
| SG14 | David | Ferreira | B003 |
| SG37 | Ana | Santos | B003 |
| SG5 | Susana | Silva | B003 |
| SL21 | Joao | Alves | B005 |
| SL41 | Julia | Borges | B005 |

Propriedade:

| idPropriedade | idFuncionario |
|---|---|
| PG4 | NULL |
| PA14 | SA9 |
| PG16 | SG14 |
| PG21 | SG37 |
| PG36 | SG37 |
| PL94 | SL41 |

```
SELECT l.idLoja, l.cidade, f.idFuncionario, pNome, uNome, idPropriedade

FROM Loja l, Funcionario f, Propriedade p

WHERE l.idLoja = f.idLoja AND f.idFuncionario = p.idFuncionario;
```

| idLoja | cidade | idFuncionario | pNome | uNome | idPropriedade |
|---|---|---|---|---|---|
| B003 | Braga | SG14 | David | Ferreira | PG16 |
| B003 | Braga | SG37 | Ana | Santos | PG21 |
| B003 | Braga | SG37 | Ana | Santos | PG36 |
| B005 | Felgueiras | SL41 | Julia | Borges | PL94 |
| B007 | Porto | SA9 | Maria | Marques | PA14 |

```
SELECT l.idLoja, l.cidade, f.idFuncionario, pNome, uNome, idPropriedade
FROM (Loja l JOIN Funcionario f USING (idLoja))
      JOIN Propriedade p USING (idFuncionario)
```

# Agrupar por múltiplas colunas

*Encontrar o número de propriedades geridas por cada funcionário*

```
SELECT f.idLoja, f.idFuncionario, COUNT(*) AS nPropriedades
FROM Funcionario f, Propriedade p
WHERE f.idFuncionario = p.idFuncionario
GROUP BY f.idLoja, f.idFuncionario;
```

| idLoja | idFuncionario | nPropriedades |
|---|---|---|
| B003 | SG14 | 1 |
| B003 | SG37 | 2 |
| B005 | SL41 | 1 |
| B007 | SA9 | 1 |

# JOINS

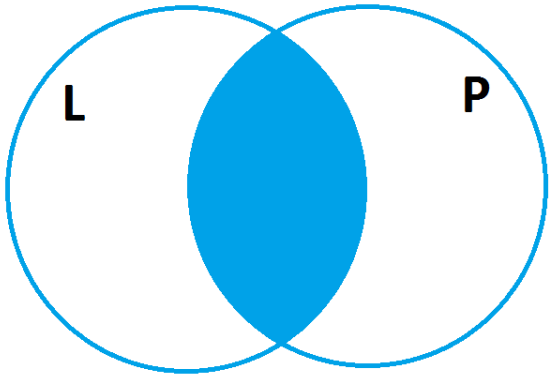
# INNER JOIN

Loja1:

| idLoja | cidade |
|---|---|
| B002 | Felgueiras |
| B003 | Braga |
| B004 | lousada |

Propriedade1 :

| idPropriedade | cidade |
|---|---|
| PG4 | Braga |
| PL94 | Felgueiras |
| PA14 | Porto |




```
SELECT l.*, p.*
FROM Loja1 l, Propriedade1 p
WHERE l.cidade = p.cidade;
```

| idLoja | cidade | idPropriedade | cidade |
|---|---|---|---|
| B003 | Braga | PG4 | Braga |
| B002 | Felgueiras | PL94 | Felgueiras |

# JOINS

# LEFT OUTER JOIN

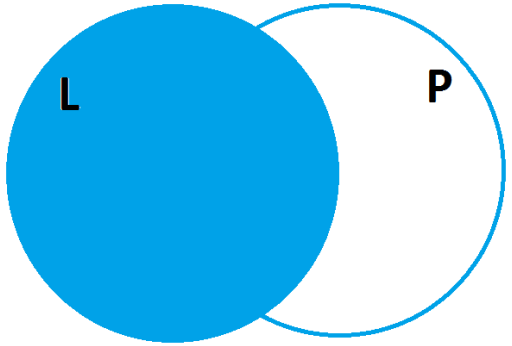
*Listrar todas as lojas e propriedades que estejam na mesma cidade*

Loja1

| idLoja | cidade |
|---|---|
| B002 | Felgueiras |
| B003 | Braga |
| B004 | lousada |

Propriedade1

| idPropriedade | cidade |
|---|---|
| PG4 | Braga |
| PL94 | Felgueiras |
| PA14 | Porto |




```
SELECT l.*, p.*
FROM Loja1 l LEFT JOIN Propriedade1 p ON l.cidade = p.cidade;
```

| idLoja | cidade | idPropriedade | cidade |
|---|---|---|---|
| B002 | Felgueiras | PL94 | Felgueiras |
| B003 | Braga | PG4 | Braga |
| B004 | lousada | NULL | NULL |

# JOINS

# RIGHT OUTER JOIN

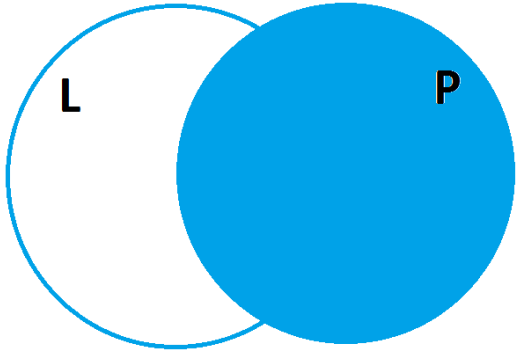
*Listar todas as propriedades e lojas que estejam na mesma cidade*

Loja1

| idLoja | cidade |
|---|---|
| B002 | Felgueiras |
| B003 | Braga |
| B004 | lousada |

Propriedade1

| idPropriedade | cidade |
|---|---|
| PG4 | Braga |
| PL94 | Felgueiras |
| PA14 | Porto |




```
SELECT l.*, p.*
FROM Loja1 l RIGHT JOIN Propriedade1 p ON l.cidade = p.cidade;
```

| idLoja | cidade | idPropriedade | cidade |
|---|---|---|---|
| B003 | Braga | PG4 | Braga |
| B002 | Felgueiras | PL94 | Felgueiras |
| NULL | NULL | PA14 | Porto |

# JOINS

# FULL OUTER JOIN

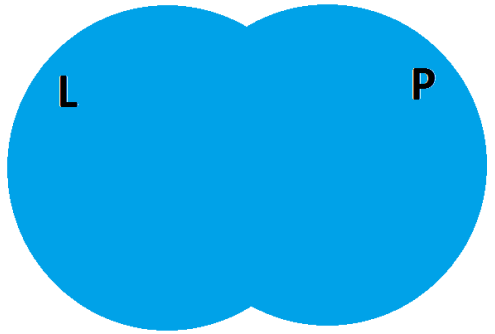
*Listar todas as lojas e propriedades que estejam na mesma cidade incluindo propriedades ou lojas desemparelhadas*

Loja1

| idLoja | cidade |
|---|---|
| B002 | Felgueiras |
| B003 | Braga |
| B004 | lousada |

Propriedade1

| idPropriedade | cidade |
|---|---|
| PG4 | Braga |
| PL94 | Felgueiras |
| PA14 | Porto |




(FULL OUTER JOIN não existe em MySQL. Pode usar-se:)

```
SELECT l.*, p.*
FROM Loja1 l LEFT JOIN Propriedade1 p ON l.cidade = p.cidade
UNION
SELECT l.*, p.*
FROM Loja1 l RIGHT JOIN Propriedade1 p ON l.cidade = p.cidade;
```

| idLoja | cidade | idPropriedade | cidade |
|---|---|---|---|
| B004 | lousada | NULL | NULL |
| B003 | Braga | PG4 | Braga |
| B002 | Felgueiras | PL94 | Felgueiras |
| NULL | NULL | PA14 | Porto |

# Utilização de EXISTS

*Encontrar todos os funcionários que trabalham na loja de Felgueiras*

```
SELECT idFuncionario, pNome, uNome, cargo
FROM Funcionario f
WHERE EXISTS (  SELECT *
                FROM Loja l
                WHERE f.idLoja = l.idLoja AND cidade='Felgueiras');
```

| idFuncionario | pNome | uNome | cargo |
|---|---|---|---|
| SL21 | Joao | Alves | Gerente |
| SL41 | Julia | Borges | Assistente |

# Combinar Resultados de tabelas

A = {1, 2, 3, 4}

B = {3, 4, 5, 6}

União: elementos que pertencem a A ou a B
- $A \cup B$ = {1, 2, 3, 4, 5, 6}

Interseção: elementos que pertencem a A e a B
- $A \cap B$ = {3, 4}

Diferença: elementos que pertencem a A mas não a B
- $A - B$ = {1, 2}

As tabelas têm de ter o mesmo número de colunas e os tipos de dados devem ser iguais

Sintaxe: `Operador [ALL][CORRESPONDING [BY {coluna1[,…]}]]` Operador = UNION, INTERSECT, EXCEPT

# UNION

*Construir uma lista de todas as cidades onde há uma loja ou uma propriedade*

```
(SELECT cidade
FROM Loja
WHERE cidade IS NOT NULL)
UNION
(SELECT cidade
FROM Propriedade
WHERE cidade IS NOT NULL);
```

Cidades em que há lojas

Cidades em que há propriedades

| cidade |
| --- |
| Felgueiras |
| Braga |
| lousada |
| Porto |

# INTERSECT

*Construir a lista de todas as cidades em que há lojas e propriedades*

(Não existe o operador em MySQL)

```
SELECT DISTINCT cidade
FROM Loja
WHERE cidade IN (SELECT cidade
                 FROM Propriedade);
```

| cidade |
| --- |
| Felgueiras |
| Braga |
| Porto |

# EXCEPT

*Construir a lista de todas as cidades em que há lojas mas não propriedades*

(Não existe o operador em MySQL)

```
SELECT DISTINCT cidade
FROM Loja
WHERE cidade NOT IN (SELECT cidade
                     FROM Propriedade);
```

| cidade |
| --- |
| lousada |

# Vistas

Resultados dinâmicos de uma ou mais operações relacionais sobre as relações base para produzir novas relações. Uma vista é uma relação virtual que não existe necessariamente na base de dados, mas pode ser produzida pelo pedido de um determinado utilizador.

```
CREATE VIEW nome [(novaColuna [, …])]
AS subselect [WITH [CASCADED | LOCAL] CHECK OPTION]
```

Eliminar:

```
DROP VIEW nome [RESTRICT | CASCADE]
```

# Vista horizontal

*Criar uma vista para que o gestor da Loja B003 possa ver apenas os detalhes dos funcionários que trabalham na sua loja*

```
CREATE VIEW gerente3Funcionarios
AS SELECT *
        FROM Funcionario
        WHERE idLoja = 'B003';
```

```
SELECT * FROM gerente3Funcionarios
```

| idFuncionario | pNome | uNome | cargo | genero | dtNascimento | salario | idLoja |
|---|---|---|---|---|---|---|---|
| SG14 | David | Ferreira | Supervisor | M | 1958-03-24 | 18000 | B003 |
| SG37 | Ana | Santos | Assistente | F | 1960-11-10 | 12000 | B003 |
| SG5 | Susana | Silva | Gerente | F | 1940-06-03 | 24000 | B003 |

# Vista Vertical

*Criar uma vista dos detalhes dos funcionários da loja B003 sem mostrar informação de salário, para que assim apenas os gerentes consigam ver o salario de quem trabalha na sua loja*

```
CREATE VIEW funcionarios3
AS SELECT idFuncionario, pNome, uNome, cargo, género
        FROM Funcionario
        WHERE idLoja = 'B003';

SELECT * FROM funcionarios3;
```

| idFuncionario | pNome | uNome | cargo | genero |
|---|---|---|---|---|
| SG14 | David | Ferreira | Supervisor | M |
| SG37 | Ana | Santos | Assistente | F |
| SG5 | Susana | Silva | Gerente | F |

# Vistas com GROUP e JOIN

*Criar uma vista dos funcionários que gerem propriedades, incluindo a loja para que trabalham, número de funcionário, e numero de propriedades que gerem*

```
CREATE VIEW FuncionarioNrPropriedades(idLoja,idFuncionario,nProp)
AS SELECT f.idLoja, f.idFuncionario, COUNT(*)
        FROM Funcionario f, Propriedade p
        WHERE f.idFuncionario = p.idFuncionario
        GROUP BY f.idLoja, f.idFuncionario;

SELECT * FROM FuncionarioNrPropriedades;
```

| idLoja | idFuncionario | nProp |
|---|---|---|
| B003 | SG14 | 1 |
| B003 | SG37 | 2 |
| B005 | SL41 | 1 |
| B007 | SA9 | 1 |

# Vistas

## VANTAGENS

- Independência de dados
- Alterações às tabelas refletem-se nas views
- Maior segurança
- Complexidade reduzida
- Conveniência
- Personalização
- Integridade dos dados

## DESVANTAGENS

- Restrições em updates
- Restrições de estrutura
- Performance