



Web
MySQL

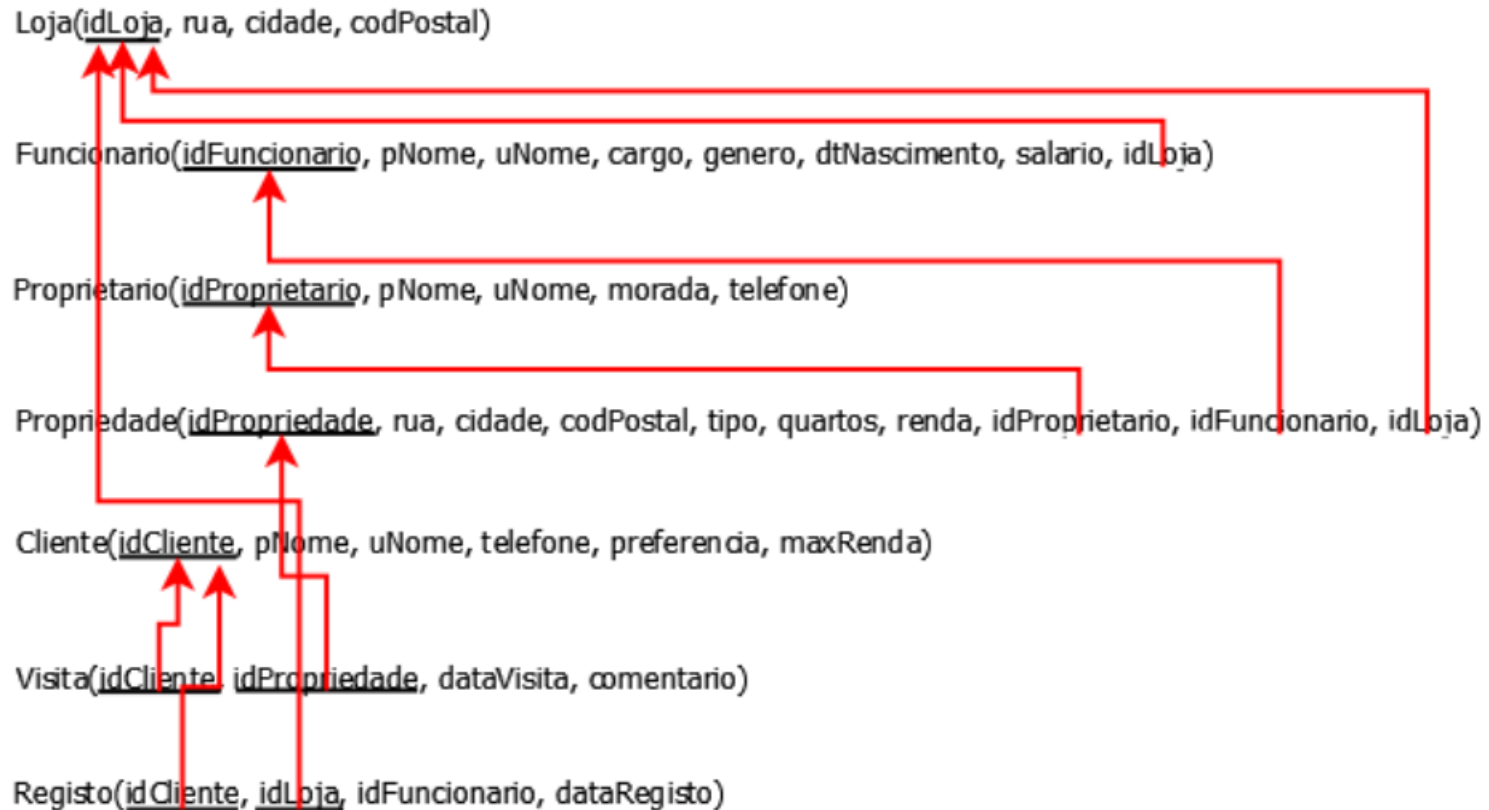
Catarina Oliveira

DCT DEPARTAMENTO **CIÊNCIA**
E TECNOLOGIA

CONTEÚDO

1. Estrutura da BD dos exemplos
2. Definição de dados
3. Tabelas
4. Modificar uma tabela
5. Seleção
6. Selecionar todas as linhas e todas as colunas
7. Selecionar colunas específicas, todas as linhas
8. DISTINCT
9. Campos Calculados
10. Seleção de linhas
10. Seleção por comparação
11. Padrões
12. Seleção por padrões
13. Seleção por NULL
14. Ordenação
15. INSERT
16. UPDATE
17. DELETE
18. Subquery com igualdade
19. Queries com múltiplas tabelas

Estrutura da BD dos exemplos



Definição de dados

Criar uma base de dados:

```
CREATE SCHEMA [nome | AUTHORIZATION idCriador]
```

```
CREATE SCHEMA imobicasa;
```

- Cria a base de dados chamada imobicasa

Eliminar uma base de dados:

```
DROP SCHEMA [RESTRICT | CASCADE]
```

```
DROP SCHEMA imobicasa;
```

- Elimina a base de dados chamada imobicasa

Tabelas

```
CREATE TABLE tabela
{(coluna tipoDados [NOT NULL] [UNIQUE]
[DEFAULT opcao] [CHECK (condicao)] [, ... ]}
[PRIMARY KEY (listOfColumns),]
{[UNIQUE (listaColunas)] [, ... ]}
{[FOREIGN KEY (listaForeignKeys)
REFERENCES tabelaForeignKey [(listaChaves)]
[MATCH {PARTIAL | FULL}
[ON UPDATE acao]
[ON DELETE acao]] [, ... ]}
{[CHECK (condicao)] [, ... ]}
```

```
DROP TABLE tabela [RESTRICT | CASCADE]
```

Exemplo:

```
CREATE TABLE Loja(
    idLoja CHAR(4) PRIMARY KEY,
    rua VARCHAR(30),
    cidade VARCHAR(30),
    codPostal VARCHAR(10)
);
```

- Cria a tabela chamada Loja, com colunas:
 - **idLoja:** Tipo CHAR(4): tem de ter exatamente 4 caracteres; chave primária
 - **rua:** Tipo VARCHAR(30): pode ter até 30 caracteres
 - **cidade:** Tipo VARCHAR(30)
 - **codPostal:** Tipo VARCHAR(10)

```
DROP TABLE Loja;
```

- Elimina a tabela Loja

Modificar uma tabela

ALTER TABLE tabela

[ADD [COLUMN] coluna tipoDados [NOT NULL] [UNIQUE]

[DEFAULT default] [CHECK (condicao)]

[DROP [COLUMN] coluna [RESTRICT | CASCADE]]

[ADD [CONSTRAINT [restricao]] defRestricao]

[DROP CONSTRAINT restricao [RESTRICT | CASCADE]]

[ALTER [COLUMN] SET DEFAULT default]

[ALTER [COLUMN] DROP DEFAULT]

Adicionar / eliminar coluna

Adicionar / eliminar restrição

Definir / eliminar valor default para coluna

Exemplos:

- Modificar a tabela Funcionario removendo o default 'Assistente' do cargo e fazendo com que o default para a coluna género seja 'F'

```
ALTER TABLE Funcionario
    ALTER cargo DROP DEFAULT;
ALTER TABLE Funcionario
    ALTER género SET DEFAULT 'F';
```

- Modificar a tabela Propriedade removendo a restrição de que os funcionários não podem gerir mais do que 100 casas. Modificar a tabela Cliente adicionando uma nova coluna que representa a preferência quanto ao número de quartos

```
ALTER TABLE Propriedade
    DROP CONSTRAINT menosDe100;
ALTER TABLE Cliente
    ADD prefNQuartos dQuartos;
```

Seleção

```
SELECT      [DISTINCT | ALL] {*} | [coluna [AS novoNome]] [, . . . ]}  
FROM        tabela [alias] [, . . . ]  
[WHERE      condicao]  
[GROUP BY   listaColunas] [HAVING condicao]  
[ORDER BY   listaColunas]
```

SELECT: especifica as colunas que devem aparecer no resultado

FROM: especifica a(s) tabela(s) a ser(em) utilizada(s)

WHERE: filtra as linhas de acordo com uma condição

GROUP BY: forma grupos de linhas com o mesmo valor numa coluna

HAVING: filtra os grupos de acordo com uma condição

ORDER BY: especifica a ordem do resultado

Selecionar todas as linhas e todas as colunas

Listar todos os detalhes de Funcionario

```
SELECT idFuncionario, pNome, uNome, cargo, genero, dtNascimento, salario, idLoja
FROM Funcionario;
```

OU

```
SELECT *
FROM Funcionario;
```

idFuncionario	pNome	uNome	cargo	genero	dtNascimento	salario	idLoja
SA9	Maria	Marques	Assistente	F	1970-02-19	9000	B007
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
SL21	Joao	Alves	Gerente	M	1945-10-01	30000	B005
SL41	Julia	Borges	Assistente	F	1965-05-13	9000	B005

Selecionar colunas específicas, todas as linhas

Produzir a lista de salários dos funcionários, mostrando apenas o número de funcionário, primeiro e último nomes e salário.

```
SELECT idFuncionario, pNome, uNome, salario  
FROM Funcionario;
```

idFuncionario	pNome	uNome	salario
SA9	Maria	Marques	9000
SG14	David	Ferreira	18000
SG37	Ana	Santos	12000
SG5	Susana	Silva	24000
SL21	Joao	Alves	30000
SL41	Julia	Borges	9000

DISTINCT

Listar os números de todas as propriedades que foram visitadas

```
SELECT idPropriedade  
FROM Visita;
```

idPropriedade
PG36
PG4
PG4
PA14
PA14



```
SELECT DISTINCT idPropriedade  
FROM Visita;
```

idPropriedade
PG36
PG4
PA14



Campos Calculados

Produzir a lista de salários mensais dos funcionários, mostrando o número de funcionário, primeiro e último nomes e salário.

```
SELECT idFuncionario, pNome, uNome, salario/12
FROM Funcionario;
```

idFuncionario	pNome	uNome	salario/12
SA9	Maria	Marques	750.0000
SG14	David	Ferreira	1500.0000
SG37	Ana	Santos	1000.0000
SG5	Susana	Silva	2000.0000
SL21	Joao	Alves	2500.0000
SL41	Julia	Borges	750.0000

```
SELECT idFuncionario, pNome, uNome, salario/12 AS salarioMensal
FROM Funcionario;
```

idFuncionario	pNome	uNome	salarioMensal
SA9	Maria	Marques	750.0000
SG14	David	Ferreira	1500.0000
SG37	Ana	Santos	1000.0000
SG5	Susana	Silva	2000.0000
SL21	Joao	Alves	2500.0000
SL41	Julia	Borges	750.0000

Seleção de linhas

Condições básicas:

- Comparação: comparar o valor de uma expressão com o de outra expressão
 - $=, <, >, !=, <=, >=$
 - AND e OR para expressões mais complexas
 - Expressão avaliada da esquerda para a direita
 - Sub-expressões entre parêntesis avaliadas primeiro
 - NOT avaliado antes de AND e OR
 - AND avaliado antes de OR
 - Recomenda-se utilização de parêntesis para evitar ambiguidades
- Intervalo: verificar se o valor de uma expressão está dentro de um determinado intervalo
- Membro: verificar se o valor de uma expressão pertence a um conjunto de valores
- Padrão: verificar se o valor de uma expressão corresponde a um determinado padrão
- Null: verificar se o valor de uma expressão é NULL

Seleção por comparação

Listar as moradas de todas as lojas no Porto ou em Braga

```
SELECT *  
FROM Loja  
WHERE cidade='Porto' OR cidade = 'Braga';
```

idLoja	rua	cidade	codPostal
B003	R. Central 34	Braga	1122
B007	Av. Aliados 2345	Porto	4321

Padrões

% representa qualquer sequência de caracteres

_ representa um caracter

LIKE 'H%' : o primeiro carater tem de ser um H, mas o resto pode ser qualquer coisa

LIKE 'H_ _ _' : a string deve ter exatamente 4 carateres e o 1º tem de ser um H

LIKE '%e' : qualquer sequência de carateres que termine num e

LIKE '%Braga%' : sequência de carateres de qualquer tamanho que contenha 'Braga'

NOT LIKE 'H%' : o primeiro carater não pode ser um H

Se a string puder conter o carater % ou _ devemos utilizar um carater de escape.

Exemplo: **LIKE '15#%'** **ESCAPE '#'** para encontrar a string '15%'

Seleção por padrões

Listar todos os proprietários que vivam em Braga

```
SELECT *  
FROM Proprietario  
WHERE morada LIKE '%Braga%';
```

idProprietario	pNome	uNome	morada	telefone
CO40	Telma	Silva	Rua Direita 123 Braga	253986758
CO87	Carolina	Marques	Rua do Meio 54 Braga	253985746
CO93	Antonio	Magalhães	Rua Principal 987 Braga	255098345

Seleção por NULL

Listar os detalhes das visitas à propriedade PG4 em que o cliente não tenha feito comentário

```
SELECT *  
FROM Visita  
WHERE idPropriedade='PG4' AND comentario IS NULL;
```

idCliente	idPropriedade	dataVisita	comentario
CR56	PG4	2005-05-26	NULL

Ordenação

Produzir uma lista de funcionários por ordem decrescente de salario

```
SELECT *  
FROM Funcionario  
ORDER BY salario DESC;
```

idFuncionario	pNome	uNome	cargo	genero	dtNascimento	salario	idLoja
SL21	Joao	Alves	Gerente	M	1945-10-01	30000	B005
SG5	Susana	Silva	Gerente	F	1940-06-03	24000	B003
SG14	David	Ferreira	Supervisor	M	1958-03-24	18000	B003
SG37	Ana	Santos	Assistente	F	1960-11-10	12000	B003
SA9	Maria	Marques	Assistente	F	1970-02-19	9000	B007
SL41	Julia	Borges	Assistente	F	1965-05-13	9000	B005

INSERT

```
INSERT INTO tabela [(listaDeColunas)]
VALUES (listadeValores)
```

Exemplo:

- Inserir uma nova linha na tabela dos funcionários fornecendo dados para todas as colunas

```
INSERT INTO Funcionario
VALUES('SG16','Andre','Brito','Assistente',
'M','1957-5-25',8300,'B003');
```

- Inserir uma nova linha na tabela dos funcionários fornecendo dados para idFuncionario, pNome, uNome, cargo, salario e idLoja

```
INSERT INTO Funcionario
(idFuncionario,pNome,uNome,cargo,salario,idLoja)
VALUES('SG16','Andre','Brito','Assitente',8300,
'B003')
```

```
INSERT INTO Funcionario
VALUES('SG16','Andre','Brito','Assitente',
NULL,NULL,8300,'B003')
```

IMP.GE.190.0

```
INSERT INTO tabela [(listaDeColunas)]
SELECT ...
```

Exemplo:

- Se tivermos uma tabela FuncionáriosGeral e quisermos inserir lá apenas o nome de cada um e a loja em que trabalha

```
INSERT INTO FuncionariosGeral
SELECT pNome, uNome, idLoja
FROM Funcionario;
```

- A query dentro do SELECT pode ser qualquer uma, desde que tenha as mesmas colunas da tabela que queremos preencher

UPDATE

```
UPDATE tabela  
SET coluna1 = valor1 [, coluna2 = valor2, ...]  
[WHERE condicao]
```

- Atualizar todas as linhas

- *Dar a todos os funcionários um aumento de 3%*

```
UPDATE Funcionario  
SET salario = salario*1.03;
```

- Atualizar linhas específicas

- *Dar a todos os gerentes um aumento de 5%*

```
UPDATE Funcionario  
SET salario = salario*1.05  
WHERE cargo = 'Gerente';
```

- Atualizar múltiplas colunas

- *Promover o funcionário 'SG14' ao cargo de gerente e subir-lhe o salario para 18000*

```
UPDATE Funcionario  
SET cargo = 'Gerente', salario=18000  
WHERE idFuncionario = 'SG14';
```

DELETE

```
DELETE FROM tabela  
[WHERE condicao]
```

- Eliminar todas as linhas
 - *Eliminar todas as linhas da tabela Visitas*

```
DELETE FROM Visitas
```

- Eliminar linhas específicas
 - *Eliminar todas as visitas que se relacionem com a propriedade 'PG4'*

```
DELETE FROM Visita  
WHERE idPropriedade = 'PG4';
```

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

Consultas encadeadas

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

Queries com múltiplas tabelas

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

Clientes:

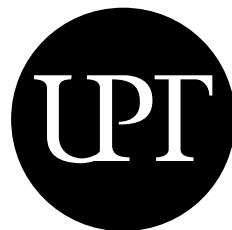
idCliente	pNome	uNome
CR56	Adelina	Santos
CR62	Maria	Pereira
CR74	Miguel	Silva
CR76	Joao	Alves

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, comentario
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



UNIVERSIDADE
PORTUCALENSE

Do conhecimento à prática.