Ficheiros ou tabelas

Conjunto de informação que caracteriza uma entidade

1 ficheiro:

1. N - registos ou tuplos
   1. Conjunto de informação que caracteriza 1 elemento da entidade
   2. N campos ou atributos
      1. Mais pequena quantidade de informação que pode ser lida ou escrita

----

Normalização de tabelas

## Regras de Normalização das tabelas

### 1ª Forma Normalizada

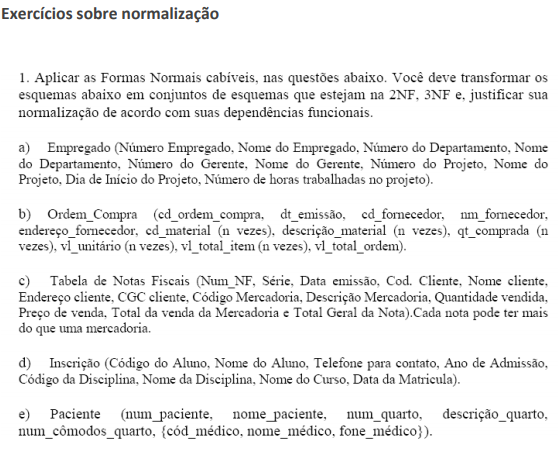
Separar os grupos repetitivos, colocar chaves primárias e relacionar as tabelas, ou seja, a chave primária da tabela será colocada na 2ª tabela

### 2ª Forma Normalizada

Verificar se os campos não-chave dependem da totalidade das chaves. Caso não existir uma dependência criar uma nova tabela

### 3ª Forma Normalizada

Analisar os campos não-chave e verificar se existe dependência entre eles, isto é se existe dependência transitiva entre os campos.



Exercicios Normalizacao

**Legenda Elenir:**

1. Sublinhado significa que é a chave da tabela
2. Cabecalho 2 - forma de normalizacao
3. Cabecalho 3 - nome da tabela

# Exercico 1a

## Nao Normalizado

### Empregado

Numero Empregado

Nome Empregado

Número Departamento

Nome Departamento

Número do Gerente

Nome do Gerente

Número do Projeto

Nome do Projeto

Dia de Início do Projeto

Numero de Horas Trabalhadas No Projecto

## 1ª Forma Normalizada

### Empregado

Numero Empregado

Nome Empregado

Número Departamento

Nome Departamento

Número do Gerente

Nome do Gerente

Número do Projeto

### Projeto

Número do Projeto

Nome do Projeto

Dia de Início do Projeto

Numero de Horas Trabalhadas No Projecto

## 2ª Forma Normalizada

### Empregado

Numero Empregado

Nome Empregado

Número Departamento

Número do Gerente

Número do Projeto

### Projeto

Número do Projeto

Nome do Projeto

Dia de Início do Projeto

Numero de Horas Trabalhadas No Projecto

### Departamento

Número Departamento

Nome Departamento

### Gerentes

Número do Gerente

Nome do Gerente

# Exercicio 1b

## Nao Normalizada

### Ordem de Compra

Cd\_ordem\_compra

Dt\_emissao

Cd\_fornecedor

Nm\_fornecedor

Endereco\_fornecedor

## 1ª Forma Normalizada

### Ordem de Compra

Cd\_ordem\_compra

Dt\_emissao

Cd\_fornecedor

Nm\_fornecedor

Endereco\_fornecedor

### Material

Cd\_material

Descricao\_material

Qt\_comprada

Vl\_unitario

Vl\_total\_item

Vl\_total\_ordem

Cd\_ordem\_compra

## 2ª Forma Normalizada

### Ordem de Compra

Cd\_ordem\_compra

Dt\_emissao

Cd\_fornecedor

Nm\_fornecedor

Endereco\_fornecedor

### Material

Cd\_material

Cd\_ordem\_compra

Qt\_comprada

Vl\_total\_ordem

### Detalhes Material

Cd\_material

Descricao\_material

VL\_unitario

## 3ª Forma Normalizada

### Material

Cd\_material

Cd\_ordem\_compra

Qt\_comprada

Vl\_total\_ordem

### Detalhes Material

Cd\_material

Descricao\_material

VL\_unitario

### Ordem de Compra

Cd\_ordem\_compra

Dt\_emissao

Vl\_total\_ordem

cd\_fornecedor

### Fornecedores

CD\_fornecedor

Nm\_fornecedor

endereco\_fornecedor

## Nao Normalizada

### Tabela de Notas Fiscais

Num\_nota\_fiscal

Serie

Data emissao

Cod cliente

Nome cliente

Endereco cliente

CGC cliente

Codigo mercadoria

Descricao mercadoria

Quantidade vendida

Preco de venda

Total da venda da mercadoria

total geral da nota (n vezes)

## 1ª Forma Normalizada

### Tabela de Notas Fiscais

Num\_nota\_fiscal

Serie

Data emissao

Cod cliente

Nome cliente

Endereco cliente

CGC cliente

total geral da nota

### Mercadorias

Codigo mercadoria

Descricao mercadoria

Quantidade vendida

Preco de venda

Total da venda da mercadoria

Num\_nota\_fiscal

## 2ª Forma Normalizada

### Tabela de Notas Fiscais

Num\_nota\_fiscal

Serie

Data emissao

Cod cliente

Nome cliente

Endereco cliente

CGC cliente

total geral da nota

### Mercadorias

Codigo mercadoria

Num\_nota\_fiscal

Quantidade vendida

Preco de venda

Total da venda da mercadoria

### Detalhes Mercadorias

Codigo mercadoria

Descricao mercadoria

## 3ª Forma Normalizada

### Detalhes Mercadorias

Codigo mercadoria

Descricao mercadoria

### Mercadorias

Codigo mercadoria

Num\_nota\_fiscal

Quantidade vendida

Preco de venda

Total da venda da mercadoria

### Tabela de Notas Fiscais

Num\_nota\_fiscal

Serie

Data emissao

Cod cliente

total geral da nota

### Clientes

Cod cliente

Nome cliente

Endereco cliente

CGC cliente

# Exercicio Quadro 1

## Nao Normalizada

### Tabela Vendedor

Codigo de vendedor

Nome vendedor

Codigo postal

Localidade

Codigo de zona

Nome de zona

### Tabela Produtos

Codigo de Produto

Descricao do Produto

Quantidade em stock

Preco Unitario

## 1 Forma Normalizada

### Tabela Vendedor

Codigo de vendedor

Nome vendedor

Codigo postal

Localidade

Codigo de zona

Nome de zona

### Tabela Produtos

Codigo de Produto

Codigo de vendedor

Descricao do Produto

Quantidade em stock

Preco Unitario

## 2 Forma Normalizada

### Tabela Vendedor

Codigo de vendedor

Nome vendedor

Codigo postal

Localidade

Codigo de zona

Nome de zona

### Tabela Produtos

Codigo de Produto

Codigo vendedor

### Tabela Detalhes do Produto

Codigo de Produto

Descricao do Produto

Quantidade em stock

Preco Unitario

## 3 Forma Normalizada

### Tabela Vendedor

Codigo de vendedor

Nome vendedor

Codigo postal

Codigo de zona

### Tabela Codigo Postal

Codigo Postal

Localidade

### Tabela Zonas

Codigo de zona

Nome de zona

### Tabela Produtos

Codigo de Produto

Codigo vendedor

### Tabela Detalhes do Produto

Codigo de Produto

Descricao do Produto

Quantidade em stock

Preco Unitario

# Exercicio 1d

## Nao Normalizada

Inscrição (Código do Aluno, Nome do Aluno, Telefone para contato, Ano de Admissão, Código da Disciplina, Nome da Disciplina, Nome do Curso, Data da Matrícula)

## 1 Forma Normalizada

### Tabela Inscricao

Código do Aluno

Nome do Aluno

Telefone para contato

Ano de Admissão

Código da Disciplina

### Tabela Disciplinas

Código da Disciplina

Nome da Disciplina

Nome do Curso

Data da Matrícula

Código do Aluno

## 2 Forma Normalizada

### Tabela Inscricao

Código do Aluno

Nome do Aluno

Telefone para contato

Ano de Admissão

### Tabela Disciplinas

Código da Disciplina

Código do Aluno

Nome do Curso

Data da Matrícula

### Tabela Detalhes de Disciplina

Codigo da disciplina

Nome Disciplina

## 3 Forma Normalizada

### Tabela Inscricao

Código do Aluno

Nome do Aluno

Telefone para contato

Ano de Admissão

### Tabela Disciplinas

Código da Disciplina

Código do Aluno

Nome do Curso

Data da Matrícula

### Tabela Detalhes de Disciplina

Codigo da disciplina

Nome Disciplina

### Trabalho Base de dados

* Introducao
* Diagrama Entidade relacao
* Normalizacao
  + 1a forma normalizada
  + 2a forma normalizada
  + 3a forma normalizada
* SQL
* Conclusao

# Exercicio 1c

## Nao Normalizada

Tabela de Notas Fiscais (Num\_NF, Serie, Data Emissao, Codigo Cliente, Nome Cliente, Endereco Cliente, CGC Cliente, Codigo Mercadoria, Descricao Mercadoria, Quantidade Vendida, Preco de venda, Total da Venda da Mercadoria e Total Geral da Nota)

## 1 Forma Normalizada

### Tabela Notas Fiscais

Num\_NF

Serie

Data Emissao

Codigo Cliente

Nome Cliente

Endereco Cliente

CGC Cliente

Total Geral da Nota

### Tabela Mercadorias

Codigo Mercadoria

Num\_NF

Descricao Mercadoria

Quantidade Vendida

Preco de venda

Total da Venda da Mercadoria

## 2 Forma Normalizada

### Tabela de Notas Fiscais

Num\_NF

Serie

Data Emissao

Codigo Cliente

Nome Cliente

Endereco Cliente

CGC Cliente

Total Geral da Nota

### Tabela Mercadorias

Codigo Mercadoria

Num\_NF

Quantidade Vendida

Preco de venda

Total da Venda da Mercadoria

### Detalhes de Mercadoria

Codigo Mercadoria

Descricao Mercadoria

## 3 Forma Normalizada

### Tabela de Notas Fiscais

Num\_NF

Serie

Data Emissao

Total Geral da Nota

Codigo Cliente

### Tabela Clientes

Codigo Cliente

Nome Cliente

Endereco Cliente

CGC Cliente

### Tabela Mercadorias

Codigo Mercadoria

Num\_NF

Quantidade Vendida

Preco de venda

Total da Venda da Mercadoria

### Detalhes de Mercadoria

Codigo Mercadoria

Descricao Mercadoria

# Exercicio 1c

## Nao Normalizada

Paciente (Numero Paciente, Nome Paciente, Numero Quarto, Descricao Quarto, Numero Mobilia, Codigo de Medico, Nome de Medico, Fone Medico)

## 1 Forma Normalizada

### Tabela Paciente

Numero Paciente

Nome Paciente

Numero Quarto

Descricao Quarto

Numero Mobilia

### Tabela Medico

Codigo de Medico

Nome de Medico

Fone Medico

Numero Paciente

## 2 Forma Normalizada

### Tabela Paciente

Numero Paciente

Nome Paciente

Numero Quarto

Descricao Quarto

Numero Mobilia

### Tabela Medico

Codigo de Medico

Numero Paciente

### Tabela Detalhes Medico

Codigo de Medico

Nome de Medico

Fone Medico

## 3 Forma Normalizada

### Tabela Paciente

Numero Paciente

Nome Paciente

Numero Quarto

### Tabela Quarto

Numero Quarto

Descricao Quarto

Numero Mobilia

### Tabela Medico

Codigo de Medico

Numero Paciente

### Tabela Detalhes Medico

Codigo de Medico

Nome de Medico

Fone Medico

Bases de dados em Microsoft Access

1. Criar a Base de dados
2. Criar as tabelas
3. Inserir a informacao
4. Criação de Relatorios
5. Consultas
6. Formulários

# Guia para criterios em Access:

* <https://support.office.com/pt-pt/article/exemplos-de-crit%C3%A9rios-de-consulta-3197228c-8684-4552-ac03-aba746fb29d8#bm2>

# Exercício Turma - Ficheiro Access

* <https://drive.google.com/open?id=1q1DOtSavJcK6X5Bcy9DKHu1TZtCtb9K9>

# Exercício Hospital - Ficheiro Access

* <https://drive.google.com/open?id=1Puz0Ows7_3cLuEe_CLBqLqIcMRVzJOll>

Bases de dados em SQL

SQL Management Studio Express

* <https://aka.ms/ssmsfullsetup>

Server: (localdb)\MSSQLLocalDB

Criacao da tabela

Create table <nome> (campos);

Insercao - Insert

Atualizacao - Update

Eliminacao

* Tabela - Drop
* Registo - Delete

Visualizar - select

Abrir a Base de dados

* Use <nome da base de dados> go

Eliminar a base de dados

Drop Database <nome>

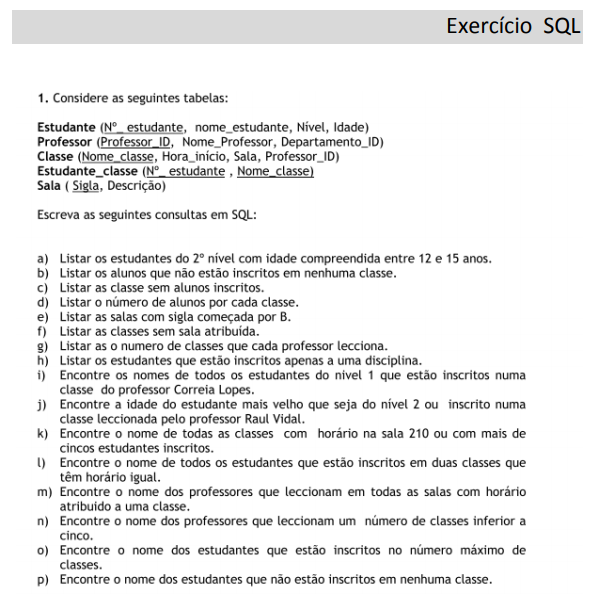
# Funcoes de agregacao

<https://www.devmedia.com.br/sql-funcoes-de-agregacao/38463>

# Codigo para criar base de dados de uma biblioteca

| ----------------------- -- Tabela CodPostais -- ----------------------- -- Codpostals (codpost1, codpost2, localidade) create table CodPostais  (  codpost1 int primary key,   codpost2 int,  localidade nvarchar(50) );  insert into dbo.CodPostais values ('4100','001','Lisboa'), ('4400','096','V.N.Gaia'), ('4000','034','Porto')  -------------------- -- Tabela Autores -- -------------------- --Autores (ida, nome, morada, codpost, datanasc) create table Autores  (  ida int primary key,   nome nvarchar(25),   morada nvarchar(25),   codPost int Foreign key references CodPostais(codPost1),  datanasc nvarchar(10) );  insert into Autores values (100,'António Manuel','Rua aeiou','4100','2010-04-15'), (120,'Joaquim Gomes','Rua xyz','4400','2010-04-14'), (123,'Maria Joaquina','Rua abc','4000','2010-04-13'), (125,'Beatriz Alves','Rua wyz','4100','2010-04-12'), (130,'Joana Afonso','Rua aaa','4400','2010-03-12')  -------------------- -- Tabela Editora -- -------------------- --Editoras (ide, nome, morada, codpost, telefone) create table Editora (  ide int primary key,  nome nvarchar(25),   morada nvarchar(25),   codPost int Foreign key references CodPostais(codPost1),  telefone int );  insert into Editora values (1,'Isla Gaia','Rua aa','4100',223775566), (2,'Porto Gaia','Rua bb','4400',223775567), (3,'Porto Lisboa','Rua cc','4000',223775568)  ------------------ -- Tabela Livro -- ------------------ -- idl, ida[->autores], ide[->editoras], título, num\_páginas, ano create table Livros (  idl int primary key,  ida int Foreign key references Autores(ida),  ide int Foreign key references Editora(ide),  titulo nvarchar(30),  num\_paginas int,  ano int )  insert into Livros values (10,100,1,'Os Maias',200,2005), (20,123,2,'Lusiadas',350,2000), (30,125,1,'Crime do Padre Amaro',100,2005), (40,123,2,'O Eclipse',500,2009)  ------------------- -- Tabela Socios -- ------------------- -- Sócios (ids, nome, morada, codpost1, datanasc, telefone) create table Socios  (   ids int primary key,  nome nvarchar(30),  morada nvarchar(25),  codPost int Foreign key references CodPostais(codPost1),  datanasc date,  telefone int ); insert into Socios values (1000,'Maria João Sousa','Rua x','4100','2000-06-30',223777755), (2000,'Maria Manuela Sousa','Rua zz','4400','2003-10-10',223334455), (3000,'Adolfo','Rua abcd','4100','1965-11-08',223747755), (4000,'Tomás','Rua aaab','4400','1975-11-08',223335567)  ------------------------ -- Tabela Requisicoes -- ------------------------ -- Requisições (idl [->livros], ids[->socios], data) create table Requisicoes  (   idl int foreign key references Livros(idl),  ids int foreign key references Socios(ids),  data date );  insert into Requisicoes values (10,1000,'2005-04-01'), (20,2000,'2005-07-03'), (10,3000,'2009-10-10'), (40,3000,'2010-03-09')  --c) Listar o nome de todos os sócios. select nome from Socios  --d) Listar o nome de todos os livros com mais de 100 páginas. select titulo, num\_paginas from Livros where num\_paginas > 100  --e) Listar, sem repetições, a localidade dos códigos postais superiores a 4000. select distinct(localidade) from CodPostais where codpost1 > 4000  --f) Listar o número total de livros. select count(\*) from livros  --g) Listar todos os livros editados em 2008 com titulo começado por S. -- insert into Livros values (50,125,1,'Segredo',100,2008) select \* from Livros where ano = 2008 and lower(titulo) like('s%')  --h) Listar todos os livros que contém a palavra SQL no titulo. --insert into Livros values (80,123,2,'Mestre em SQL - Bases de Dados',250,2003) select \* from Livros where lower(titulo) like('%sql%')  --i) Número de requisições feitas em 2005. select count(\*) as numero\_req from Requisicoes where YEAR(data) = 2005  --j) Nomes de sócios residentes em Lisboa. select nome, localidade from Socios, CodPostais where localidade = 'Lisboa' AND Socios.codPost = CodPostais.codPost1   --k) Títulos dos livros cujo número de páginas é superior à média. select titulo from Livros where num\_paginas > (select AVG(num\_paginas) from Livros)  --l) Títulos dos livros requisitados pelo sócio nº 123. select titulo from Requisicoes, Livros where ids = 1000 and Requisicoes.idl = Livros.idl  --m) Lista de nomes de editoras e títulos dos livros por elas editados. select nome, titulo from Editora, Livros where Editora.ide = Livros.ide  --n) Lista do número de sócios e do número de requisições feitas por cada um. select ids, count(\*) as numero\_req from Requisicoes group by ids  --o) Nomes de autores que editam livros de mais do que uma editora. select nome from Autores, Livros where Autores.ida = Livros.ida group by Livros.ida, nome having count(ide) > 1  --p) Lista de todas as requisições efectuadas em 2005 ordenadas pelo número de sócio e incluindo as datas e os títulos dos livros requisitados. select Requisicoes.ids, Livros.ano, Livros.titulo  from Requisicoes, Livros  where YEAR(Requisicoes.data) = 2005 and Livros.idl = Requisicoes.idl  order by Requisicoes.ids  --q) Visualizar os socios que nao tem requisicoes  select ids from Socios where ids not in (select ids from Requisicoes) |
| --- |
|  |
|  |

Exercicio da aula



Dados:

ESTUDANTE

(100, 'João', 2, 12)

(101, 'Maria', 1, 14)

(102, 'Sinésio', 1, 16)

(103, 'Joaquina', 3, 12)

PROFESSOR

(10, 'António Gomes', 20)

(11, 'Arnaldo Silva', 20)

(12, 'Sónia Costa', 22)

(13, 'Ana Gonçalves', 23)

CLASSES

('a1', '12:00', 'aa1', 10)

('b1', '10:00', 'bb1', 11)

('c1', '14:00', 'cc1', 12)

('d1', '16:00','dd1', 11)

SALAS

('aa1', 'informatica')

('bb1', 'Quimica')

('cc1', 'Análise')

('dd1', 'técnicas')

ESTUDANTE\_CLASSE

(100, 'a1')

(103, 'b1')

(102, 'a1')

(103, 'c1')

BASE DE DADOS FABRICA 1 (Daniel)

| -- criacao de base de dados -- CREATE DATABASE fabrica  -- TABELA CODIGOS\_POSTAIS create table Codigos\_Postais (  cod\_postal nvarchar(8) primary key not null,   localidade nvarchar(30) not null ); -- INSERIR CODIGOS POSTAIS1 insert into Codigos\_Postais values ('4460-280','Barbeitos'), ('2799-554','Linda-a-Velha'), ('7370-170', 'Campo Maior'), ('4100-001','Lisboa'), ('4400-096','V.N.Gaia'), ('4000-034','Porto');  -- TABELA CLIENTES create table Clientes( cod\_cli int primary key not null, nome\_cli nvarchar(35) not null, morada\_cli nvarchar(35) null, cod\_postal nvarchar(8) foreign key references Codigos\_Postais(cod\_postal) null, contacto\_cli int null, nif\_cli int null ); -- INSERIR CLIENTES insert into Clientes values   (1,'Diogo Ramiro,'Rua de Alverca','4100-001', 225695555, 235265585),  (2,'Jaime Delgado','Rua das violetas','4400-096', 935262352, 265851455), (3,'Artur Lima','Rua das fontainhas','4000-034', 965232255, 265851520), (4,'Gerson Magno','Rua das fontainhas','4000-034', 965232255, 265851520);  -- TABELA PRODUTOS create table Produtos( cod\_prod int primary key not null, nome\_prod nvarchar(35) not null, descricao\_prod nvarchar(50) null, peso\_prod decimal (16, 2) null, preco\_prod money default 0, quantidade\_prod int null );  -- CRIAR PRODUTOS insert into Produtos values (1, 'Delta', 'Capsulas de Cafe', 0.50, 0.20, 300000), (2, 'Buondi', 'Capsulas de Cafe', 0.55, 0.25, 200000), (3, 'Nespresso', 'Capsulas de Cafe', 0.55, 0.25, 100000), (4, 'Dulce Gusto', 'Capsulas de Cafe', 0.55, 0.30, 500000);  -- TABELA ENCOMENDAS create table Encomendas( cod\_enc int primary key not null, data\_enc datetime null, quantidade\_prod int null, preco\_enc money default 0 ); -- INSERIR ENCOMENDAS insert into Encomendas values (1, '2019-01-01 10:30:00', 100, 20), (2, '2020-01-02 11:30:00', 200, 50), (3, '2020-01-03 12:30:00', 300, 75), (4, '2020-01-04 13:30:00', 1000, 250);  -- TABELA FORNECEDORES create table Fornecedores( cod\_forn int primary key not null, nome\_forn nvarchar(35) not null, morada\_cli nvarchar(35) null, cod\_postal nvarchar(8) foreign key references Codigos\_Postais(cod\_postal) null, contacto\_forn int null, ); -- INSERIR FORNECEDORES insert into Fornecedores values (1, 'Delta Cafe', 'Campo Maior Rua B, nº 4', '7370-170', 268009200), (2, 'Nestlé Portugal SA', 'R. Alexandre Herculano 8', '2799-554', 808200644);  -- TABELA FORNECEDORES create table Prod\_Forn( cod\_forn int foreign key references Fornecedores(cod\_forn) not null, cod\_prod int foreign key references Produtos(cod\_prod) not null, ); -- INSERIR RELACAO ENTRE FORNECEDORES-PRODUTOS insert into Prod\_Forn values (1, 1), (2, 2), (2, 3), (2, 4);  -- TABELA FORNECEDORES create table Enc\_Prod\_Cli( cod\_cli int foreign key references Clientes(cod\_cli) not null, cod\_prod int foreign key references Produtos(cod\_prod) not null, cod\_enc int foreign key references Encomendas(cod\_enc) not null );  -- INSERIR RELACAO ENTRE ENCOMENDAS-PRODUTOS-CLIENTES  (1, 1, 1), (2, 2, 2), (3, 3, 3), (3, 4, 4);   -- Listar o nome de todos os clientes. select nome\_cli from Clientes  -- Listar o nome dos produtos com quantidade superior a 150.000 select nome\_prod, quantidade\_prod from Produtos where quantidade\_prod > 150000  -- Listar, sem repetições, a localidade dos códigos postais superiores a 5000. select distinct(localidade) from Codigos\_Postais where cod\_postal > '5000'  -- Listar o número total de Produtos e associar a tabela 'total\_prod'. select count(\*) as total\_prod from Produtos  -- Listar todos os produtos com quantidade superior a 120000 e o nome termina em 'o' select \* from Produtos where quantidade\_prod > 120000 and lower(nome\_prod) like('%o')  -- Listar todos os clientes cuja morada contém a palavra 'das'. select \* from Clientes where lower(morada\_cli) like('%das%')  -- Número de encomendas feitas em 2020. select count(\*) as numero\_req from Encomendas where YEAR(data\_enc) = 2020  -- Nomes de clientes residentes no Porto. select nome\_cli, localidade from Clientes, Codigos\_Postais where localidade = 'Porto' AND Clientes.cod\_postal = Codigos\_Postais.cod\_postal   -- Qual o preco da Quantidade de produto superior à média. select preco\_enc from Encomendas where quantidade\_prod > (select AVG(quantidade\_prod) from Encomendas)  -- Nomes dos produtos encomendados pelo cliente nº 3. select nome\_prod from Produtos, Enc\_Prod\_Cli where cod\_cli = 3 and Enc\_Prod\_Cli.cod\_prod = Produtos.cod\_prod  -- Lista de fornecedoes e produtos por eles fornecidos. select nome\_forn, nome\_prod from Produtos, Fornecedores, Prod\_Forn  where Fornecedores.cod\_forn = Prod\_Forn.cod\_forn and Produtos.cod\_prod = Prod\_Forn.cod\_prod  -- Lista do número de encomendas feitas for cada cliente. select nome\_cli, count(\*) as numero\_enc from Encomendas, Enc\_Prod\_Cli, Clientes  where Enc\_Prod\_Cli.cod\_cli = Clientes.cod\_cli and Encomendas.cod\_enc = Enc\_Prod\_Cli.cod\_enc group by nome\_cli  -- Nomes de clientes que fizeram mais do que uma encomenda. select nome\_cli from Clientes, Enc\_Prod\_Cli where Clientes.cod\_cli = Enc\_Prod\_Cli.cod\_cli group by Clientes.cod\_cli, nome\_cli having count(cod\_enc) > 1  -- Lista de todas as encomendas efectuadas em 2020 ordenadas pelo número de cliente e incluindo as datas e o nome dos produtos encomendados. select Clientes.cod\_cli, nome\_cli, Encomendas.data\_enc, Produtos.nome\_prod  from Clientes, Encomendas, Produtos, Enc\_Prod\_Cli where YEAR(Encomendas.data\_enc) = 2020  and Encomendas.cod\_enc = Enc\_Prod\_Cli.cod\_enc  and Produtos.cod\_prod = Enc\_Prod\_Cli.cod\_prod  and Clientes.cod\_cli = Enc\_Prod\_Cli.cod\_cli order by Clientes.cod\_cli |
| --- |

BASES DE DADOS II

int(11) = 2 elevado a 11bytes

Dominio:

* Conjunto de valores que o atributo pode assumir

Entidade:

* definem uma tabela.
* Conjunto de atributos (primarios, externos) nao pode ter valores nulos

Cardinalidade

* Um para um
* Um para muitos
* Muitos para muitos -- N:M
  + Possivelmente tem de ser convertido em 1 para muitos ou muitos para 1

Chave estrangeira - 1 valor para cada atributo

Uniao depende do esquema da relacao

## Dependencias Funcionais Exemplos

Data de Encomenda → N. Encomenda

Cliente → N. Cliente

Endereco → Cliente

Produto → Cod Produto

Quantidade → N. Encomenda

<https://demo.phpmyadmin.net/master-config/>

<https://mariadb.com/kb/en/mysql-command-line-client/>

mysql Commands

There are also a number of commands that can be run inside the client. Note that all text commands must be first on line and end with ';'

Command Deion

?, \? Synonym for `help'.

clear, \c Clear the current input statement.

connect, \r Reconnect to the server. Optional arguments are db and host.

delimiter, \d Set statement delimiter.

edit, \e Edit command with $EDITOR.

ego, \G Send command to mysql server, display result vertically.

exit, \q Exit mysql. Same as quit.

go, \g Send command to mysql server.

help, \h Display this help.

nopager, \n Disable pager, print to stdout.

notee, \t Don't write into outfile.

pager, \P Set PAGER [to\_pager]. Print the query results via PAGER.

print, \p Print current command.

prompt, \R Change your mysql prompt. See prompt command for options.

quit, \q Quit mysql.

rehash, \# Rebuild completion hash.

source, \. Execute an SQL file. Takes a file name as an argument.

status, \s Get status information from the server.

system, \! Execute a system shell command. Only works in Unix-like systems.

tee, \T Set outfile [to\_outfile]. Append everything into given outfile.

use, \u Use another database. Takes database name as argument.

charset, \C Switch to another charset. Might be needed for processing binlog

with multi-byte charsets.

warnings, \W Show warnings after every statement.

nowarning, \w Don't show warnings after every statement.

---------------------------------------------------------------------------------------------

---------------------------------------------------------------------------------------------

CREATE database db1 character set latin1 collate latin1\_swedish\_ci;

CREATE TABLE IF NOT EXISTS postal (

Codigo int(4) NOT NULL,

Local char(30) NOT NULL,

PRIMARY KEY (Codigo)

) ENGINE=InnoDB;

INSERT INTO postal (Codigo, Local)

VALUES

('1000', 'LISBOA'),

('1100', 'LISBOA'),

('1200', 'LISBOA'),

('1500', 'LISBOA'),

('2000', 'SANTAREM'),

('2300', 'TOMAR'),

('3000', 'COIMBRA'),

('4000', 'PORTO'),

('9000', 'FUNCHAL');

-

Select \* from postal where local='Lisboa' ;

-

CREATE TABLE IF NOT EXISTS pessoa (

Id int(4) NOT NULL,

Nome char(30) NOT NULL,

Idade date NOT NULL,

Salario decimal(10,2) NOT NULL,

Telefone char(12) DEFAULT NULL,

Cod\_Postal int(4) NOT NULL,

PRIMARY KEY (Id),

KEY Cod\_Postal (Cod\_Postal)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

INSERT INTO pessoa (Id, Nome, Idade, Salario, Telefone, Cod\_Postal)

VALUES

('5', 'Célia Morais', '1970-03-10', '170000.00', '123456', '1100'),

('14', 'Nascimento Augusto', '1974-02-21', '220000.00', '456123', '2300'),

('25', 'Paulo Viegas', '1966-10-31', '95000.00', NULL, '1500'),

('32', 'Florinda Simões', '1991-08-11', '147000.00', NULL, '4000'),

('37', 'Isabel Espada', '1982-07-14', '86000.00', NULL, '1100'),

('42', 'António Dias', '1992-11-12', '74000.00', '789654', '1500'),

('49', 'José António', '1995-06-30', '210000.00', NULL, '1500');

-

CREATE TABLE IF NOT EXISTS mensagem (

Id\_Msg int(2) NOT NULL,

Mensagem char(30) NOT NULL,

PRIMARY KEY (Id\_Msg)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

INSERT INTO mensagem (Id\_Msg, Mensagem)

VALUES

('10', 'Comissão de Vendas'),

('20', 'Fretes Individuais'),

('30', 'Fretes Empresas'),

('40', 'Vendas Extra'),

('50', 'Deslocações'),

('60', 'Refeições'),

('70', 'Combustíveis'),

('80', 'Transportes'),

('90', 'Telefonemas'),

('100', 'Ofertas');

-

2^(4)\*2 = 32

Unsigned = nao vai usar o sinal - 1 bit - se não tem números negativos nao vai usar o sinal

For example, if an int typically holds values from -32768 to 32767, an unsigned int will hold values from 0 to 65535.

* \_000 0001 = 1
* -000 0001 = -1

255.255.255.255 - 32 bytes

255.0.0.0 - 8 bytes

mysql.exe -h localhost -u root

https://demo.phpmyadmin.net/master-config/

https://mariadb.com/kb/en/mysql-command-line-client/

mysql Commands:

There are also a number of commands that can be run inside the client. Note that all text commands

must be first on line and end with ';'

Command Definition

?, \? Synonym for `help'.

clear, \c Clear the current input statement. -- VERY HANDY

connect, \r Reconnect to the server. Optional arguments are db and host.

delimiter, \d Set statement delimiter.

edit, \e Edit command with $EDITOR.

ego, \G Send command to mysql server, display result vertically.

exit, \q Exit mysql. Same as quit.

go, \g Send command to mysql server.

help, \h Display this help.

nopager, \n Disable pager, print to stdout.

notee, \t Don't write into outfile.

pager, \P Set PAGER [to\_pager]. Print the query results via PAGER.

print, \p Print current command.

prompt, \R Change your mysql prompt. See prompt command for options.

quit, \q Quit mysql.

rehash, \# Rebuild completion hash.

source, \. Execute an SQL file. Takes a file name as an argument.

status, \s Get status information from the server.

system, \! Execute a system shell command. Only works in Unix-like systems.

tee, \T Set outfile [to\_outfile]. Append everything into given outfile.

use, \u Use another database. Takes database name as argument.

charset, \C Switch to another charset. Might be needed for processing binlog

with multi-byte charsets.

warnings, \W Show warnings after every statement.

nowarning, \w Don't show warnings after every statement.

---------------------------------------------------------------------------------------------

https://mariadb.com/kb/en/connect-data-types/

https://mariadb.com/kb/en/format/

---------------------------------------------------------------------------------------------

CREATE TABLE IF NOT EXISTS comissao (

Id int (2) NOT NULL,

Id\_Msg int(2) NOT NULL,

Valor decimal(10,2) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

INSERT INTO comissao (Id, Id\_Msg, Valor)

VALUES

('14', '10', '10500'),

('25', '10', '2500'),

('14', '100', '3750'),

('14', '70', '400'),

('37', '40', '20'),

('37', '30', '14230'),

('37', '10', '5500'),

('14', '60', '2600'),

('25', '30', '370'),

('40', '20', '20'),

('37', '50', '120'),

('42', '20', '170'),

('49', '20', '2300');

--

CREATE TABLE IF NOT EXISTS mensagem (

Id\_Msg tinyint UNSIGNED NOT NULL,

Mensagem char(30) NOT NULL,

PRIMARY KEY (Id\_Msg)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

INSERT INTO mensagem (Id\_Msg, Mensagem)

VALUES

(10, 'Comissão de Vendas'),

(20, 'Fretes Individuais'),

(30, 'Fretes Empresas'),

(40, 'Vendas Extra'),

(50, 'Deslocações'),

(60, 'Refeições'),

(70, 'Combustíveis'),

(80, 'Transportes'),

(90, 'Telefonemas'),

(100, 'Ofertas');

--

CREATE TABLE IF NOT EXISTS funcionario (

Id smallint UNSIGNED NOT NULL,

Nome char(30) NOT NULL,

data\_nasc date NOT NULL,

Salario decimal(10,2) NOT NULL,

Telefone char(13) DEFAULT NULL,

Cod\_Postal mediumint UNSIGNED NOT NULL,

PRIMARY KEY (Id),

KEY Cod\_Postal (Cod\_Postal)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

INSERT INTO funcionario (Id, Nome, data\_nasc, Salario, Telefone, Cod\_Postal)

VALUES

('5', 'Célia Morais', '1995-10-20', '170000.00', '+35191657106', '1100'),

('14', 'Nascimento Augusto','1985-01-09', '220000.00', '+35191657987', '2300'),

('25', 'Paulo Viegas', '1987-06-30', '95000.00', NULL, '1500'),

('32', 'Florinda Simões', '1960-12-20', '147000.00', NULL, '4000'),

('37', 'Isabel Espada', '1988-07-25', '86000.00', NULL, '1100'),

('42', 'António Dias', '1970-04-05', '74000.00', '+35193657199', '1500'),

('49', 'José António', '2003-04-04', '210000.00', NULL, '1500');

--

CREATE TABLE IF NOT EXISTS postal (

Codigo mediumint NOT NULL,

Local char(30) NOT NULL,

PRIMARY KEY (Codigo)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

INSERT INTO postal (Codigo, Local)

VALUES

('1000', 'LISBOA'),

('1100', 'LISBOA'),

('1200', 'LISBOA'),

('1500', 'LISBOA'),

('2000', 'SANTAREM'),

('2300', 'TOMAR'),

('3000', 'COIMBRA'),

('4000', 'PORTO'),

('9000', 'FUNCHAL');

--

CREATE TABLE IF NOT EXISTS postal (

Codigo mediumint UNSIGNED NOT NULL,

Local char(30) NOT NULL,

PRIMARY KEY (Codigo)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

INSERT INTO postal (Codigo, Local)

VALUES

('1000', 'LISBOA'),

('1100', 'LISBOA'),

('1200', 'LISBOA'),

('1500', 'LISBOA'),

('2000', 'SANTAREM'),

('2300', 'TOMAR'),

('3000', 'COIMBRA'),

('4000', 'PORTO'),

('9000', 'FUNCHAL');

--

ALTER TABLE funcionario

ADD CONSTRAINT funcionario\_ibfk\_1 FOREIGN KEY (Cod\_Postal)

REFERENCES postal (Codigo);

-----------------------------------------------------------------------------------------------

select id, nome, data\_nasc, telefone, local

from funcionario, postal

where cod\_postal=codigo;

---

-

select nome AS Nome, @IDADE := Year(CURDATE())-Year(f.data\_nasc) AS Idade, @IDADE+1 AS "Idade Proximo Ano"

from funcionario f

order by 3;

select nome, (Year(CURDATE())-Year(f.data\_nasc)) AS idade from funcionario as f where idade > 20 ;

select nome, @IDADE:= (Year(CURDATE())-Year(f.data\_nasc)) AS `idade` from funcionario as f having 'idade' < 49 ;