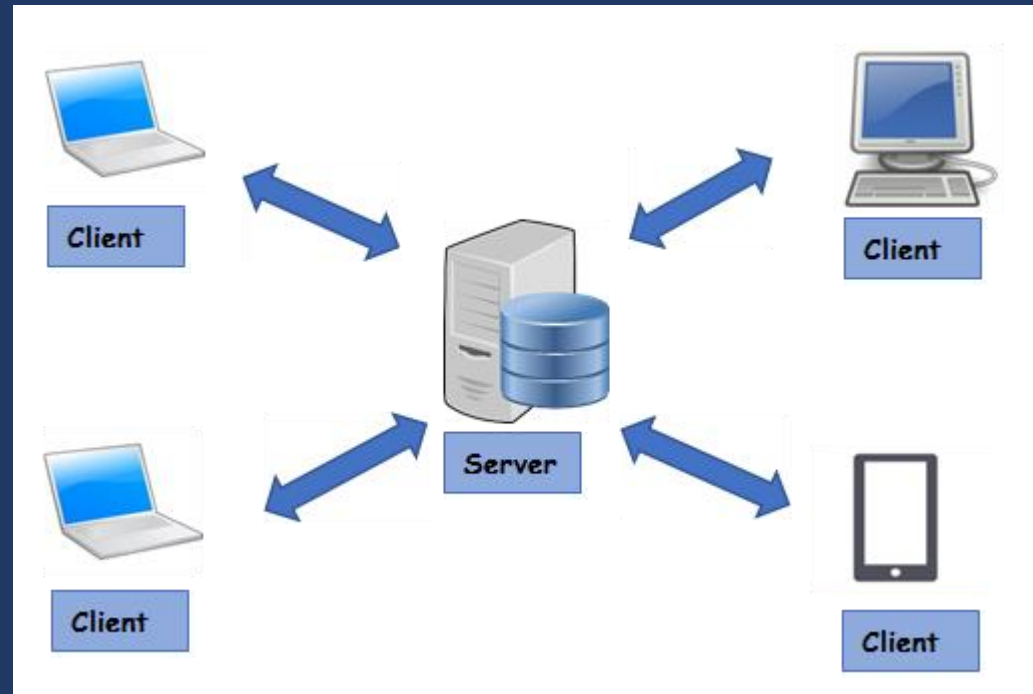


# Arquitetura cliente - servidor



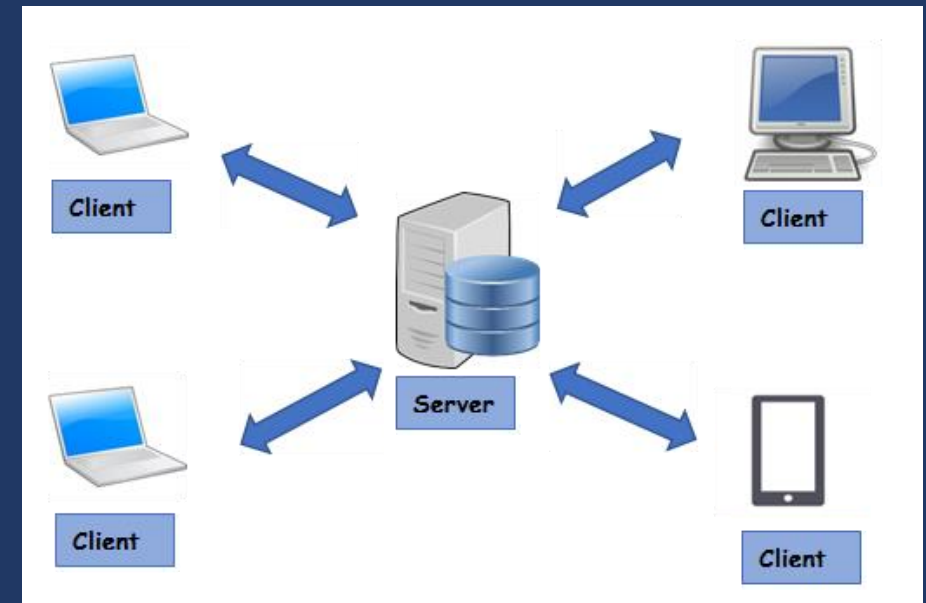
# Arquitetura cliente - servidor

Objectivo: processar informação de uma forma distribuída.

Sistema com um servidor que fornece e gere recursos e serviços a serem consumidos pelo(s) cliente(s).

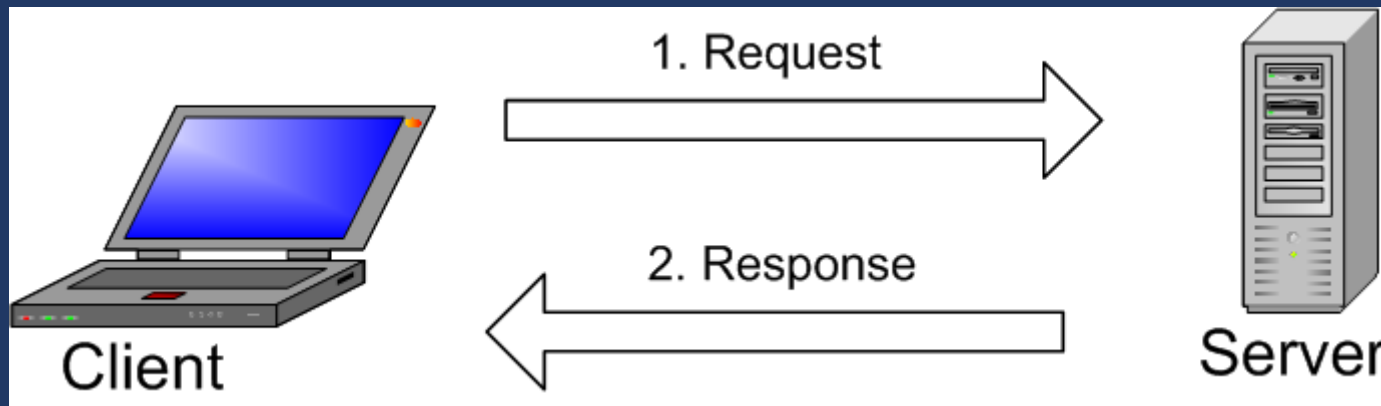
Nesta arquitectura, os vários clientes estão ligados ao servidor através de uma rede ou da Internet.

Os recursos fornecidos são partilhados.



# Arquitectura cliente - servidor

O cliente efectua o pedido do recurso (ou serviço) ao servidor através da rede, e o servidor responde fornecendo o mesmo.



O servidor, com maior capacidade de processamento e/ou armazenamento, pode fornecer a vários clientes em simultâneo e o cliente pode estar ligado a vários servidores também em simultâneo.



# Arquitetura cliente - servidor

Cliente (que pode ser visto como um processo front-end):

- 1) pede ao servidor recursos / serviços e recebe o acesso aos mesmos
- 2) tem tipicamente uma interface gráfica para o utilizador (GUI)
- 3) Pode ser um simples computador ou outro dispositivo com capacidade para pedir serviços/recursos.

Servidor (que pode ser visto como um processo – back-end):

- 1) fornece recursos / serviços
- 2) continuamente em execução e à espera de pedidos
- 3) pode “servir” vários clientes em simultâneo
- 4) com maior capacidade de processamento
- 5) Tem um endereço IP e está tipicamente à escuta numa determinada porta
- 6) pode ele próprio ser cliente



# Arquitectura cliente - servidor

Características:

- Acesso tem que ser transparente (o cliente não precisa saber a localização dos recursos)
- Independente do SO, aplicação e hardware do cliente.
- Arquitectura tem que ser escalável horizontalmente (mais clientes) e verticalmente (mais capacidade dos servidores)



# Servidor de Bases de Dados

É um servidor que guarda e gere dados para uma rede de utilizadores e/ou equipamentos.

O servidor de bases de dados está no centro do fluxo de informação de qualquer organização.

Tem que ser garantida:

Inter-relação entre dados

Integridade dos dados: objectivo é evitar mudanças imprevistas dos dados

Redundância não controlada dos dados: informação repetida no sistema (Pode ser controlada - sistema distribuído)

Para evitar redundância não controlada → partilha de informação



# Servidor de Bases de Dados

Uma base de dados é informação que pode ser facilmente acedida, gerida e actualizada.

Pode ser vista como uma conjunto de dados organizados e relacionados entre eles.

Tipicamente é gerida por um servidor.

Uma base de dados relacional é uma base de dados baseada no modelo relacional, ou seja, em tabelas relacionadas entre si.



# Bases de dados relacionais

Numa base de dados relacional, cada linha da tabela é um registo com uma identificação única, chamada chave primária. Cada coluna corresponde a um atributo, e cada registo tem tipicamente um valor para cada um deles.

Customers					
	CustomerId	FirstName	LastName	DateCreated	Client
+	1	Homer	Simpson	13/06/2014 3:33:37 PM	
+	2	Peter	Griffin	13/06/2014 9:09:56 PM	
+	3	Stewie	Griffin	13/06/2014 9:16:07 PM	
+	4	Brian	Griffin	13/06/2014 9:16:36 PM	
+	5	Cosmo	Kramer	13/06/2014 9:16:41 PM	
+	6	Philip	Fry	13/06/2014 9:17:02 PM	
+	7	Amy	Wong	13/06/2014 9:22:05 PM	
+	8	Hubert J.	Farnsworth	13/06/2014 9:22:19 PM	
+	9	Marge	Simpson	13/06/2014 9:22:37 PM	
+	10	Bender	Rodríguez	13/06/2014 9:22:52 PM	
+	11	Turanga	Leela	13/06/2014 9:23:37 PM	
*	(New)			15/06/2014 9:00:01 PM	

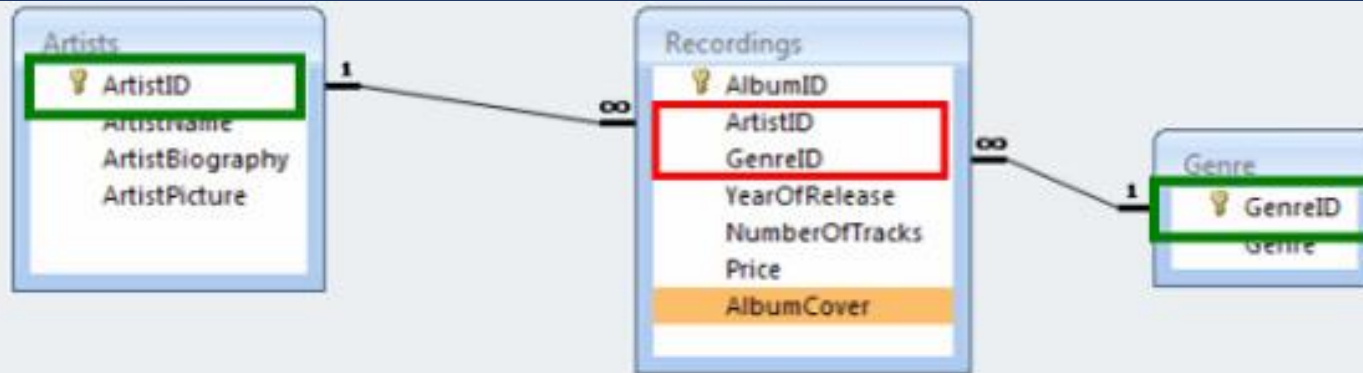




# Chave primária e chave estrangeira

Uma chave primária (primary key) identifica um registo. Uma tabela só pode ter uma chave primária que é não nula e única.

Uma chave estrangeira é um campo numa tabela que se refere a uma chave primária de outra tabela (foreign key).



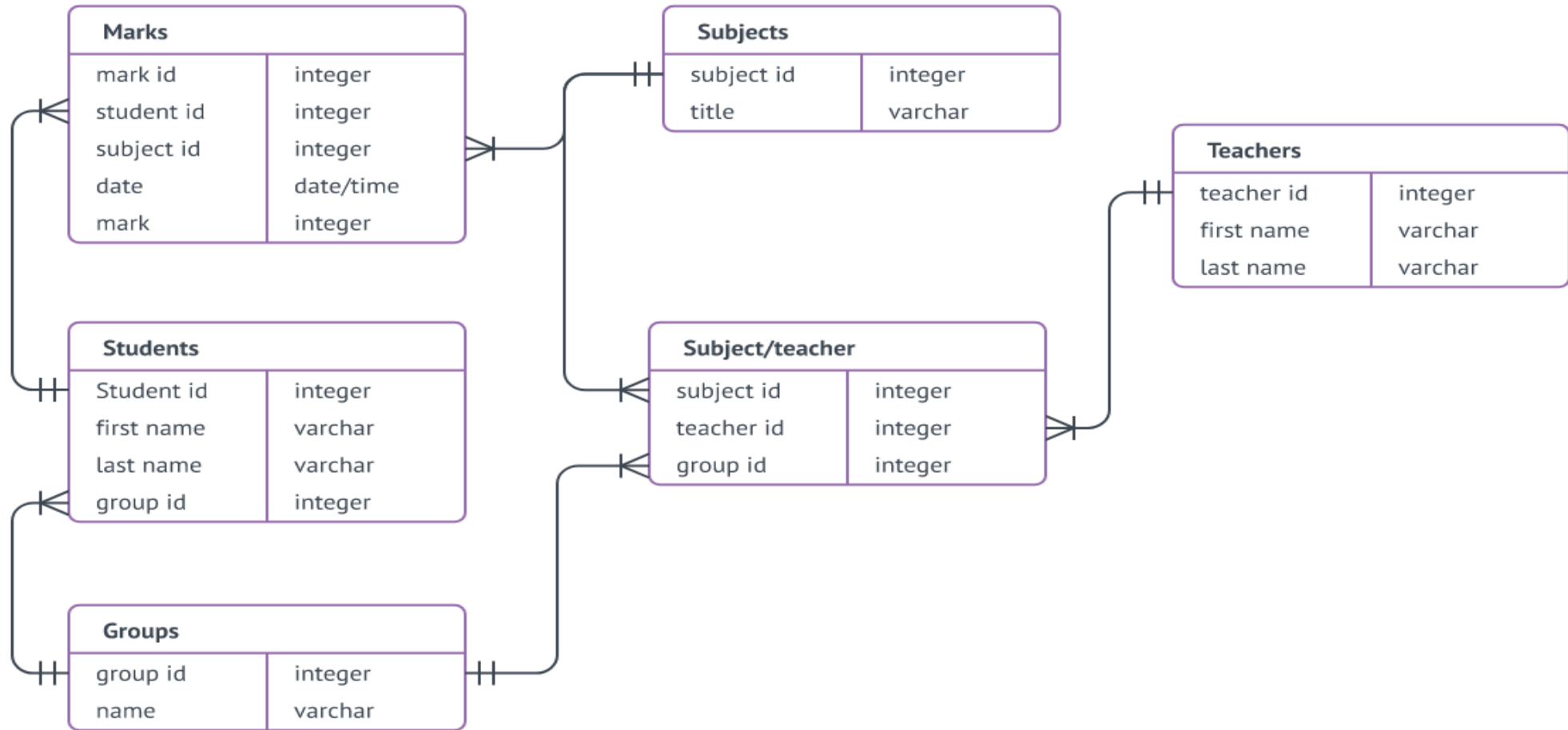
= primary key



= foreign key

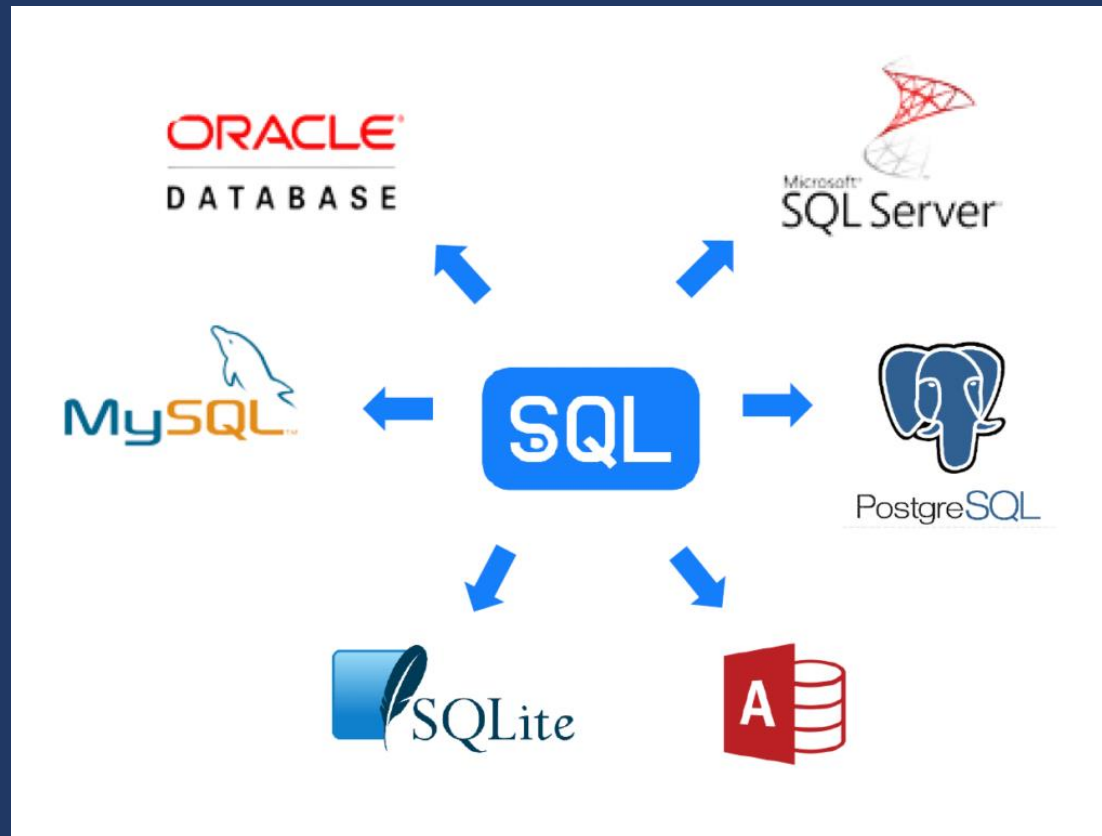


# Diagrama ER



# Servidor de Bases de Dados - SQL

O SQL é a linguagem usada para gerir uma base de dados relacional.



# Servidor de Bases de Dados - SQL

MySQL - Principais instruções:

- CREATE DATABASE – para criar uma base de dados
- CREATE TABLE – para criar uma tabela
- INSERT INTO – para inserir registos numa tabela
- UPDATE SET- para actualizar registos numa tabela
- SELECT - para extrair informação de uma ou várias tabelas



# Servidor de Bases de Dados - SQL

- CREATE DATABASE cria uma base de dados:

CREATE DATABASE nomeDaBaseDedados

Cuidado: não usar nomes compostos !



# Servidor de Bases de Dados - SQL

- CREATE TABLE cria uma tabela dentro de uma base de dados:

```
CREATE TABLE nomeDaBaseDeDadosJaExistente.nomeDaTabela (  
Campo1 tipo(dígitos) PRIMARY KEY/NULL/NOT NULL DEFAULT xxx,  
...  
)
```

Tipos disponíveis: int, varchar, tinyint, smallint, int, date, texto, bigint, decimal, float, double, real, boolean, etc.

O nome e o tipo do campo são obrigatórios.

Os restantes atributos são opcionais.

Não pode haver duas tabelas com o mesmo nome na mesma base de dados.



# Servidor de Bases de Dados - SQL

- INSERT INTO insere registos numa tabela:

```
INSERT INTO nomeDaBaseDedados.tabela(campo1,campo2,...)  
VALUES (valor1,valor2),(valo3,valor4),...
```

Se inserir em todos os campos:

```
INSERT INTO nomeDaBaseDedados.tabela  
VALUES (valor1,valor2),(valo3,valor4),...
```



# Servidor de Bases de Dados - SQL

- UPDATE SET- altere um ou vários registos de uma tabela, seleccionados através de WHERE:

```
UPDATE nomeDaBaseDedados.tabela  
SET campo1=valor1  
WHERE campo2=valor2
```

```
ou UPDATE nomeDaBaseDedados.tabela  
SET campo1=valor1  
WHERE campo2 LIKE 'a%j_'
```

```
UPDATE nomeDaBaseDedados.tabela  
SET campo1=valor1  
WHERE campo2 LIKE valor2 OR campo3 LIKE ...
```

```
ou UPDATE nomeDaBaseDedados.tabela  
SET campo1=valor1  
WHERE campo2 LIKE valor2 AND campo3>valor3
```





# Servidor de Bases de Dados - SQL

- SELECT obtém registos de uma tabela:

```
SELECT campo1,campo2,... FROM nomeDaBaseDedados.tabela  
WHERE condição1 OR/AND condição2 OR/AND ...
```

A condição pode ser uma comparação de textos: campo1='exemplo' / campo1 LIKE 'exemplo' / campo1 NOT LIKE 'ex%p\_o' (% indica 0 a infinitos caracteres, e \_ indica 1 caracter).

Ou uma comparação algébrica: campo2>3 / campo2=3 / campo2<3

Para seleccionar todos os campos:

```
SELECT * FROM nomeDaBaseDedados.tabela  
WHERE ...
```



# Servidor de Bases de Dados - SQL

- SELECT:

```
SELECT campo1,campo2,... FROM nomeDaBaseDedados.tabela  
WHERE condição1 OR/AND condição2 OR/AND ...  
ORDER BY campo1 ASC, campo2, DEC, campo3 ASC
```

ORDER BY ordena os resultados num ou vários níveis

