

Aplicações - SQL

Banco de Dados: Teoria e Prática

André Santanchè e Patrícia Cavoto
Instituto de Computação - UNICAMP
Setembro de 2016



Linguagens de Query

- Para manipulação e recuperação de dados
- Linguagens de Query (LQ) em BD:
 - Fundamentação formal
 - Subsidiária otimização
- LQ \leftrightarrow linguagens de programação
 - não se espera que sejam “Turing completas”.
 - não pensadas para cálculos complexos.
 - suportam acessos simples e eficientes a extensos conjuntos de dados

(Ramakrishnan, 2003)

Linguagens de

Uma linguagem é dita “Turing completa” se puder ser demonstrado que ela é computacionalmente equivalente à máquina de Turing.

- Para manipulação e recuperação
- Linguagens de Query (LQ) em BD:
 - Fundamentação formal
 - Subsidiária otimização
- LQ <> linguagens de programação
 - não se espera que sejam “Turing completas”.
 - não pensadas para cálculos complexos.
 - suportam acessos simples e eficientes a extensos conjuntos de dados

(Ramakrishnan, 2003)

SQL

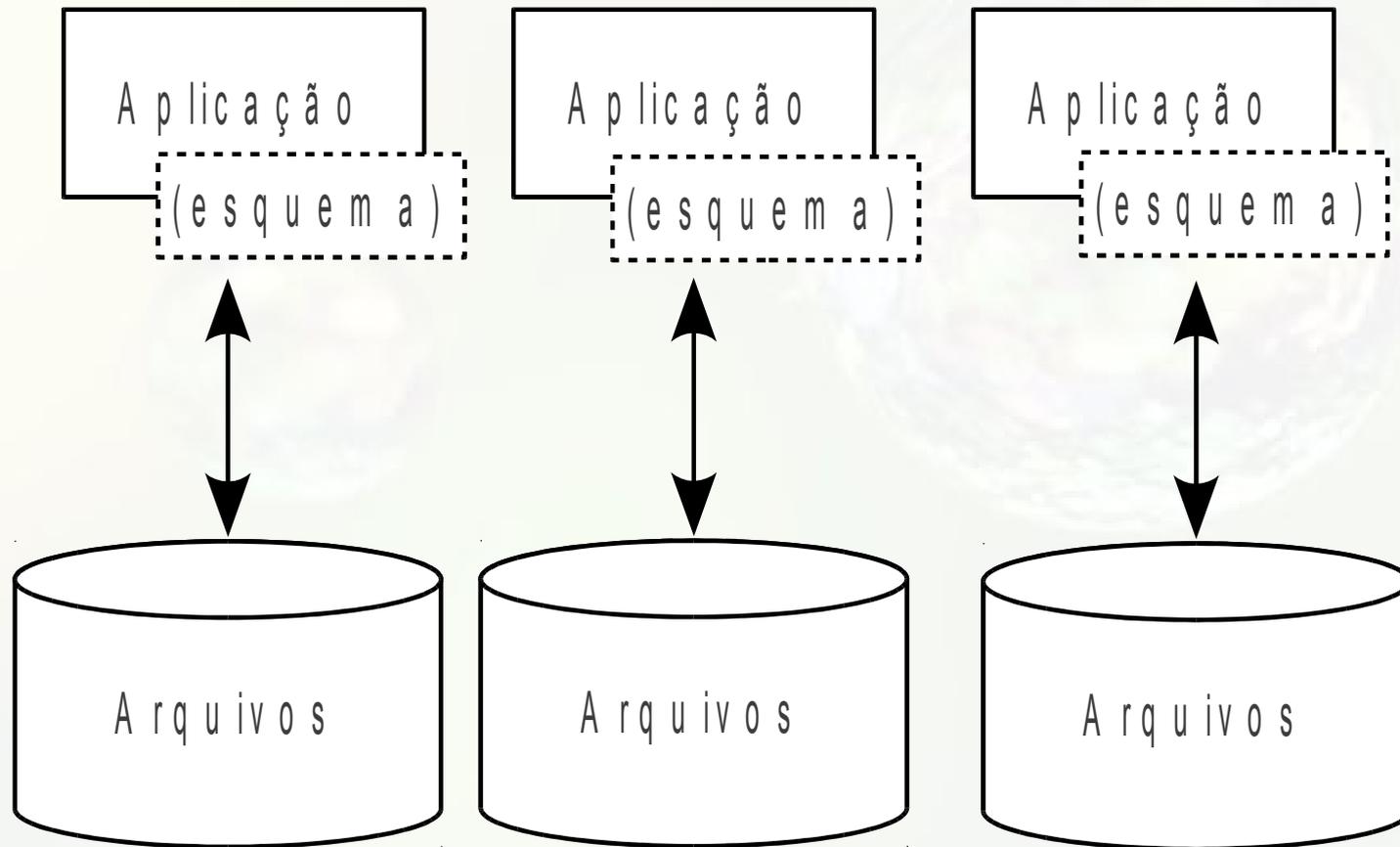
- SQL - Structured Query Language
- Originalmente: SEQUEL - Structured English QUERy Language
- Criada pela IBM Research
 - Interface BD Relacional → SYSTEM R

SQL Padronização

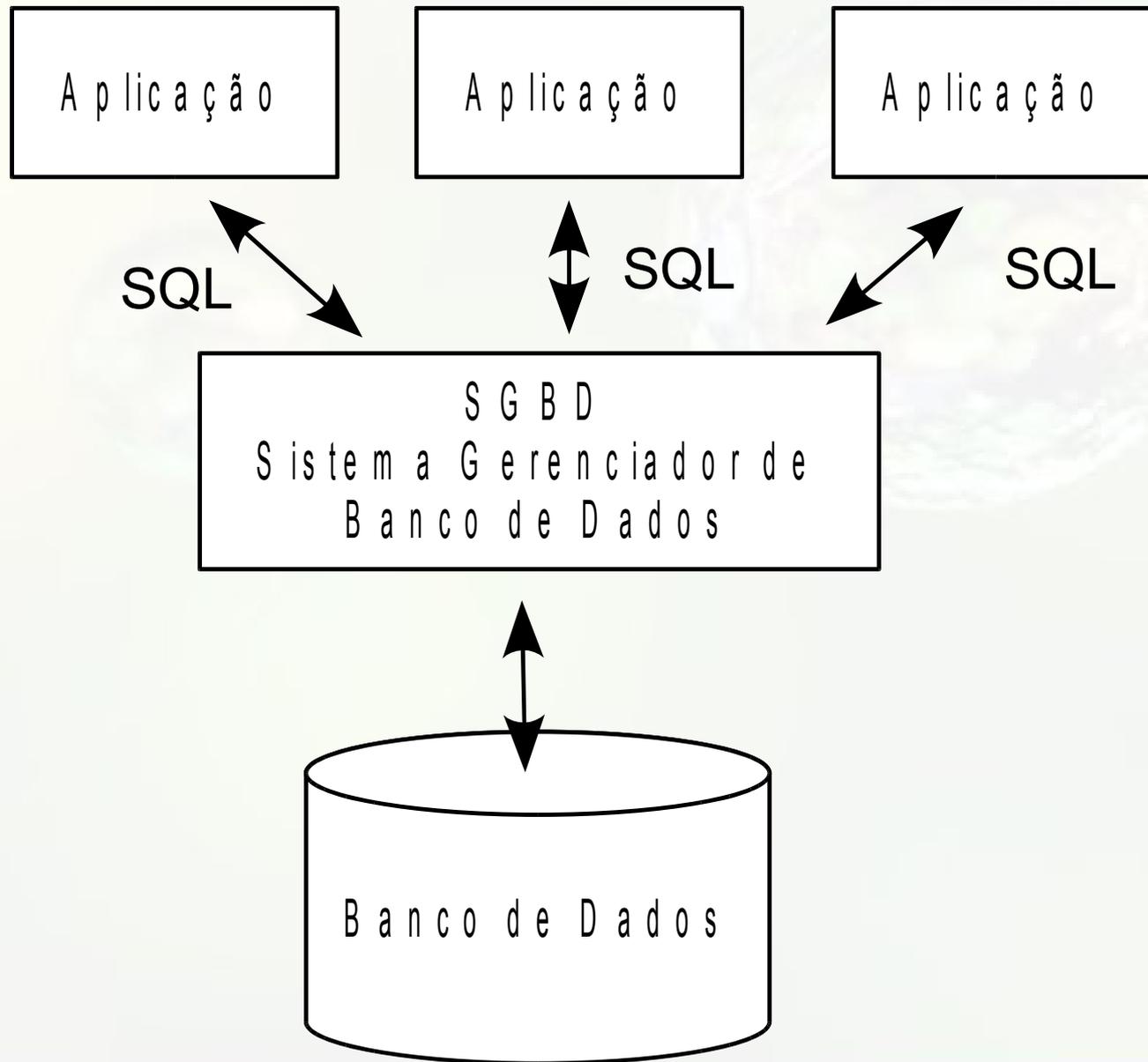
- ANSI + ISO
- SQL-86 ou SQL1
- SQL-92 ou SQL2
- SQL:1999 ou SQL3
- SQL:2003
- SQL:2006

Aplicações e Armazenamento

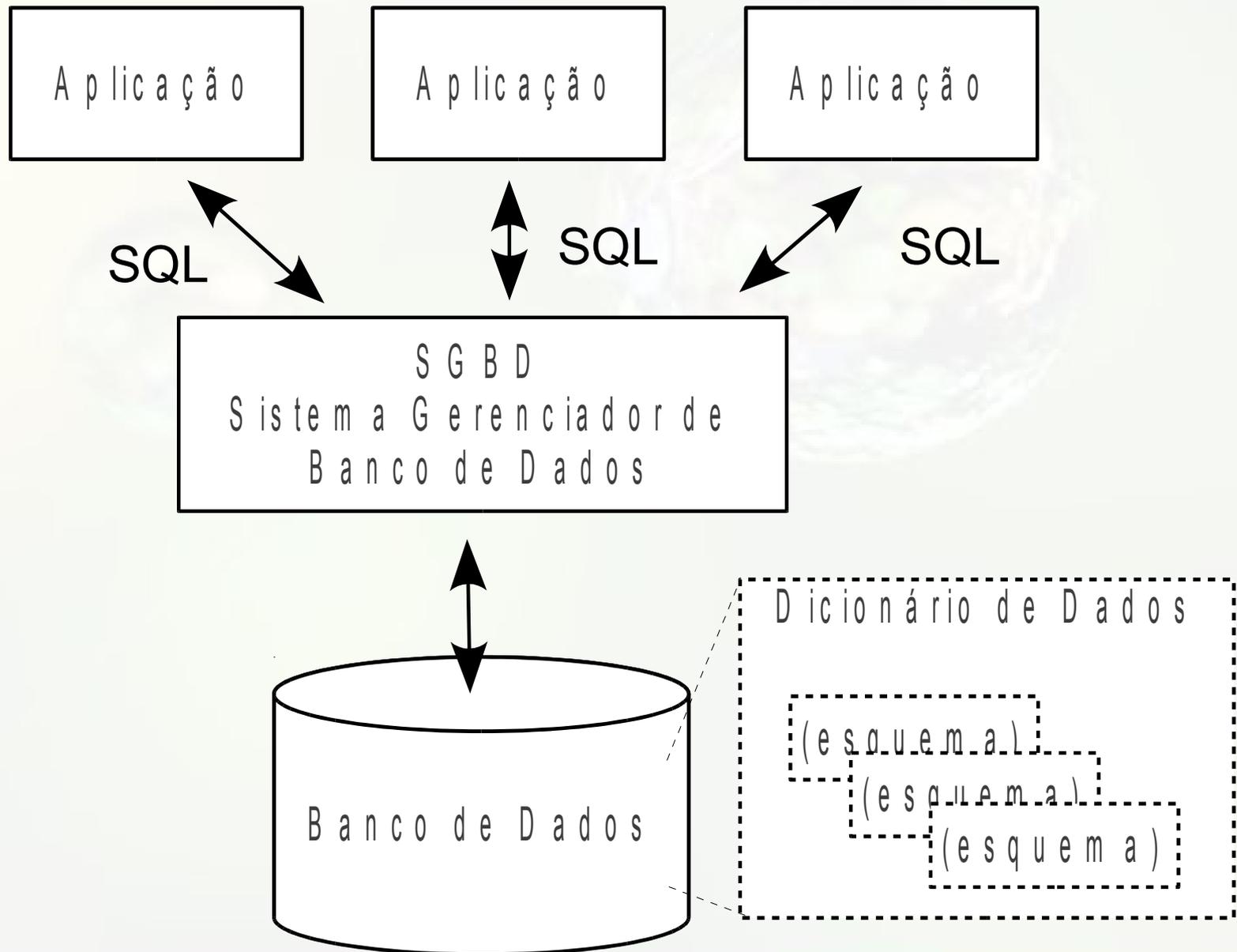
Arquivos

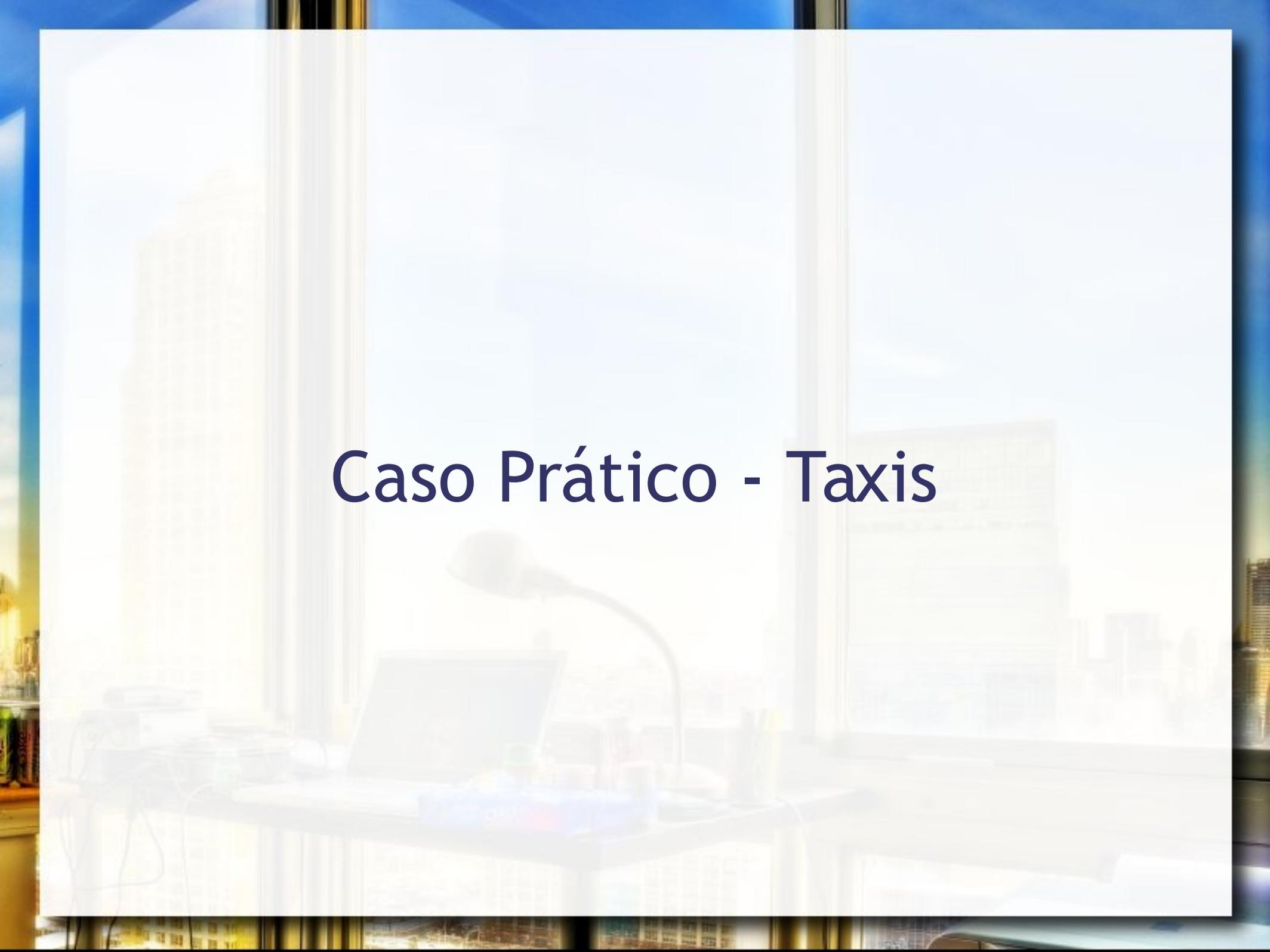


Aplicações e Armazenamento SGBD



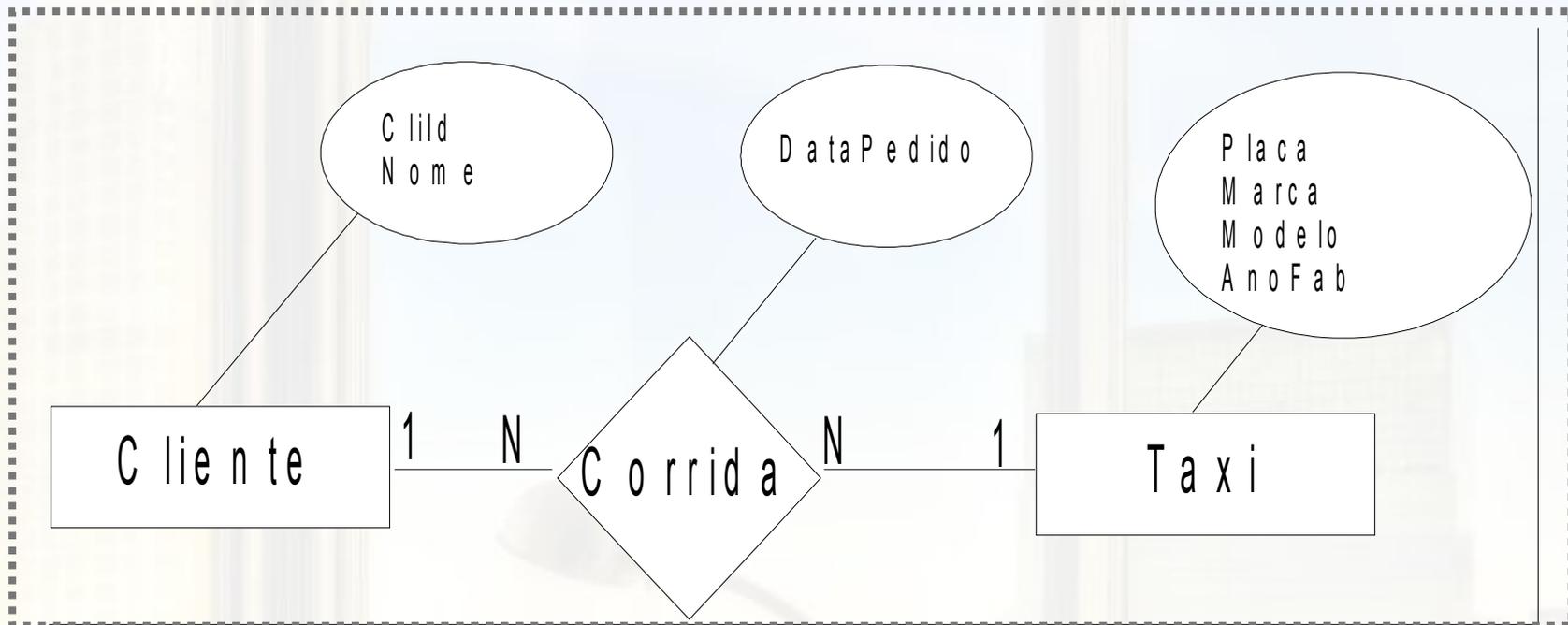
Dicionário de Dados





Caso Prático - Taxis

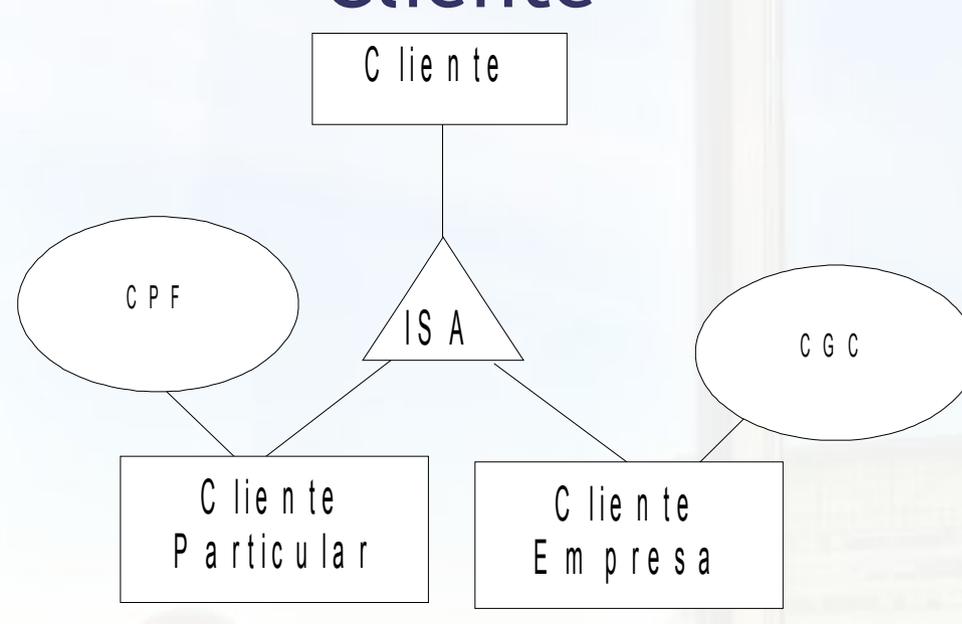
Esquema Conceitual - Exemplo Táxis



Este é um subconjunto do Estudo de Caso proposto “Despacho e controle de Táxis via terminais móveis ligados on-line com um sistema multi-usuário” por prof. Geovane Cayres Magalhães

Esquema Conceitual - Exemplo

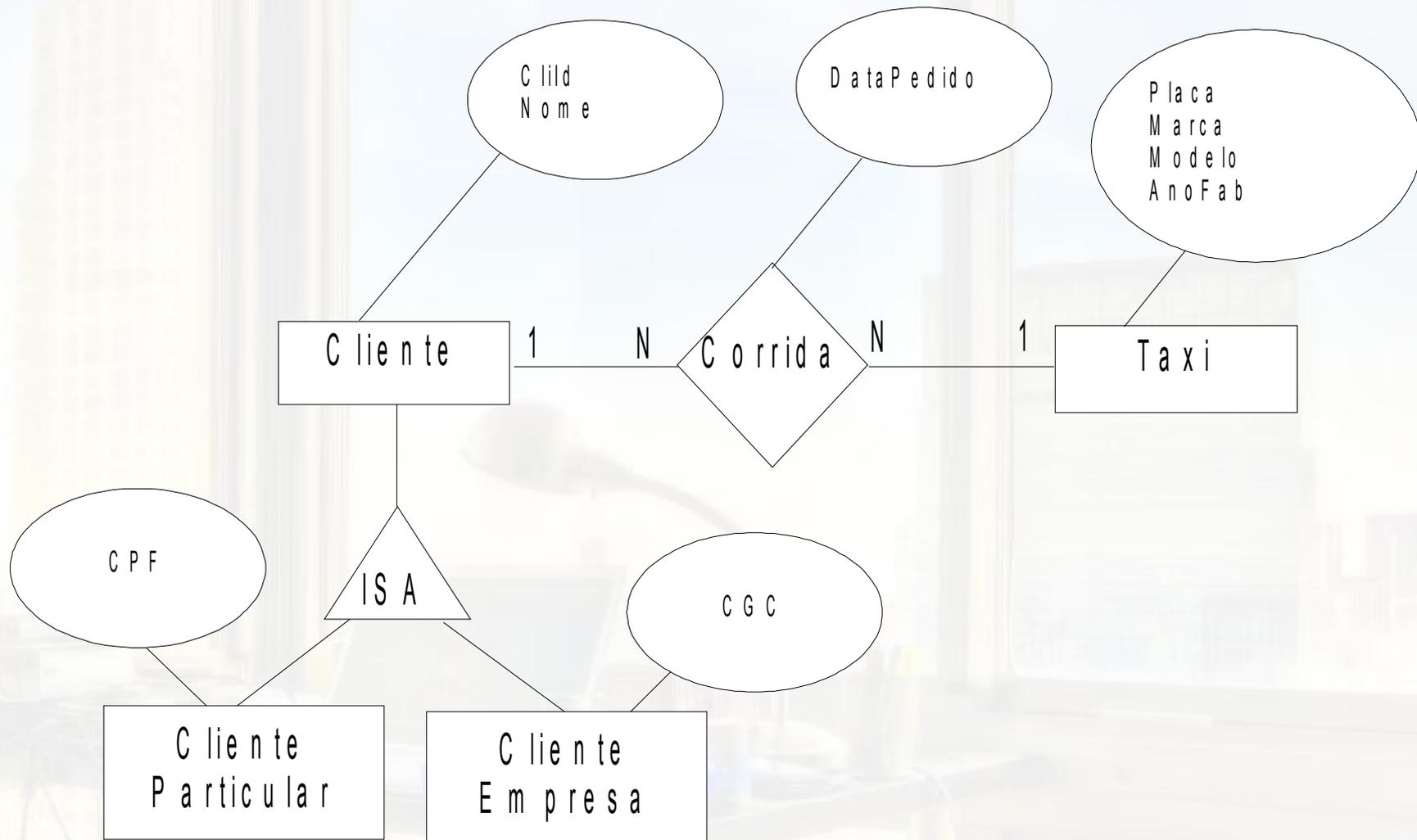
Cliente



Para ilustrar o tema apresentado, foram acrescentadas duas entidades que são especialização de Cliente. A primeira representa um indivíduo que irá pagar a conta, a segunda representa um funcionário de uma empresa conveniada, para a qual a conta será enviada. Um cliente pode pertencer a ambas especializações.

Esquema Conceitual completo

Táxis



Tabelas para exemplo - Táxis

Cliente Particular (CP)

<u>C liId</u>	N o m e	C P F
1 5 3 2	A s d r ú b a l	4 4 8 . 7 5 4 . 2 5 3 - 6 5
1 7 5 5	D o r i a n a	5 6 7 . 3 8 7 . 3 8 7 - 4 4
1 7 8 0	Q u i n c a s	5 4 6 . 3 7 3 . 7 6 2 - 0 2

Cliente Empresa (CE)

<u>C liId</u>	N o m e	C G C
1 5 3 2	A s d r ú b a l	7 5 4 . 8 5 6 . 9 6 5 / 0 0 0 1 - 5 4
1 6 4 4	J e p e t o	4 7 8 . 6 5 2 . 6 3 5 / 0 0 0 1 - 7 5
1 7 8 0	Q u i n c a s	5 5 4 . 6 6 3 . 9 9 6 / 0 0 0 1 - 8 7
1 9 8 2	Z a n d o r	7 3 6 . 9 5 2 . 3 6 9 / 0 0 0 1 - 2 3



Tabelas para exemplo - Táxis

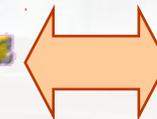
Táxi (TX)

<u>Placa</u>	<u>Marca</u>	<u>Modelo</u>	<u>AnoFab</u>
D A E 6 5 3 4	F o r d	F i e s t a	1 9 9 9
D K L 4 5 9 8	W o l k s v a g e n	G o l	2 0 0 1
D K L 7 8 7 8	F o r d	F i e s t a	2 0 0 1
J D M 8 7 7 6	W o l k s v a g e n	S a n t a n a	2 0 0 2
J J M 3 6 9 2	C h e v r o l e t	C o r s a	1 9 9 9



Corrida (R1)

<u>Ciid</u>	<u>Placa</u>	<u>DataPedido</u>
1 7 5 5	D A E 6 5 3 4	1 5 / 0 2 / 2 0 0 3
1 9 8 2	J D M 8 7 7 6	1 8 / 0 2 / 2 0 0 3



CREATE SCHEMA

- `CREATE SCHEMA <esquema>`
`AUTHORIZATION <id_autorizado>`
- Java: `executeUpdate(...)`

CREATE TABLE

- **CREATE TABLE <tabela>**
(<campo₁> <tipo> [NULL|NOT NULL] [restrição],
[...,
<campo_n> <tipo> [NULL|NOT NULL] [restrição],
PRIMARY KEY <chave_primaria>])
- **Java: executeUpdate(...)**

CREATE TABLE

```
CREATE TABLE Taxi (  
  Placa VARCHAR(7) NOT NULL,  
  Marca VARCHAR(30) NOT NULL,  
  Modelo VARCHAR(30) NOT NULL,  
  AnoFab INTEGER,  
  Licenca VARCHAR(9),  
  PRIMARY KEY(Placa)  
);
```

```
CREATE TABLE Cliente (  
  CliId VARCHAR(4) NOT NULL,  
  Nome VARCHAR(80) NOT NULL,  
  CPF VARCHAR(14) NOT NULL,  
  PRIMARY KEY(CliId)  
);
```

CREATE TABLE FOREIGN KEY

- CREATE TABLE <tabela>

...

FOREIGN KEY (<coluna_estr>₁[,...,<coluna_estr>_n])

REFERENCES <tabela_ref>([<coluna_ref>[,...,<coluna_ref]])

[ON DELETE <ação_ref>]

[ON UPDATE <ação_ref>]

- <ação_ref>

- NO ACTION → impede a ação na tabela mestre <tabela_ref>
- CASCADE → propaga a ação da tabela mestre
- SET NULL → valores de referências alterados para nulo
- SET DEFAULT → valores de referências alterados para default

CREATE TABLE FOREIGN KEY

```
CREATE TABLE Corrida (  
  CliId VARCHAR(4) NOT NULL,  
  Placa VARCHAR(7) NOT NULL,  
  DataPedido DATE NOT NULL,  
  PRIMARY KEY (CliId, Placa, DataPedido),  
  FOREIGN KEY (CliId)  
    REFERENCES Cliente (CliId)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  FOREIGN KEY (Placa)  
    REFERENCES Taxi (Placa)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION  
);
```

Exercício 1

- Escreva um comando SQL para criar os esquemas:
 - Pessoa(nome, nome_da_mãe, ano_nascimento, nome_cidade_natal)
 - nome_cidade_natal → CHE Cidade
 - Cidade(nome_cidade, sigla_estado)

INSERT

- INSERT INTO <tabela>
[(<campo₁>[,..., <campo_n>])]
VALUES (<valor₁>[,..., <valor_n>])
- executeUpdate(...)

INSERT

```
INSERT INTO Cliente  
VALUES ('1755', 'Doriana', '567.387.387-44');
```

```
INSERT INTO Taxi  
VALUES ('DAE6534', 'Ford', 'Fiesta', 1999, 'MN572345');
```

```
INSERT INTO Corrida  
VALUES ('1755', 'DAE6534', '2003-02-15');
```

Exercício 2

- Escreva um comando SQL para inserir uma tupla na tabela Pessoa com os seus dados e dados de familiares próximos (cerca de 2 linhas). Preencha a tabela Cidade com as cidades listadas na tabela Pessoa e suas respectivas siglas de estado. Use dados fictícios se preciso.

SELECT

- `SELECT * | <campo1>[,..., <campon>]
FROM <tabela1>[,..., <tabelan>]
WHERE <condição/junção>`
- `executeQuery(...)`

SELECT Projeção

SELECT Marca, Modelo FROM Taxi

<u>Placa</u>	Marca	Modelo	Ano Fab
D A E 6 5 3 4	Ford	Fiesta	1 9 9 9
D K L 4 5 9 8	Wolkswagen	Gol	2 0 0 1
D K L 7 8 7 8	Ford	Fiesta	2 0 0 1
J D M 8 7 7 6	Wolkswagen	Santana	2 0 0 2
J J M 3 6 9 2	Chevrolet	Corsa	1 9 9 9

SELECT Projeção

SELECT **Marca**, **Modelo** FROM Taxi

<u>Placa</u>	Marca	Modelo	AnoFab
D A E 6 5 3 4	Ford	Fiesta	1 9 9 9
D K L 4 5 9 8	Wolksvagen	Gol	2 0 0 1
D K L 7 8 7 8	Ford	Fiesta	2 0 0 1
J D M 8 7 7 6	Wolksvagen	Santana	2 0 0 2
J J M 3 6 9 2	Chevrolet	Corsa	1 9 9 9

SELECT Projeção

SELECT **Marca**, **Modelo** FROM Taxi

M a r c a	M o d e l o
F o r d	F i e s t a
W o l k s v a g e n	G o l
F o r d	F i e s t a
W o l k s v a g e n	S a n t a n a
C h e v r o l e t	C o r s a

SELECT Seleção

SELECT * FROM Taxi WHERE AnoFab > 2000

<u>Placa</u>	Marca	Modelo	AnoFab
D A E 6 5 3 4	Ford	Fiesta	1999
D K L 4 5 9 8	Wolksvagen	Gol	2001
D K L 7 8 7 8	Ford	Fiesta	2001
J D M 8 7 7 6	Wolksvagen	Santana	2002
J J M 3 6 9 2	Chevrolet	Corsa	1999

SELECT Seleção

SELECT * FROM Taxi WHERE AnoFab > 2000

<u>Placa</u>	Marca	Modelo	AnoFab
DAE6534	Ford	Fiesta	1999
DKL4598	Wolksvagen	Gol	2001
DKL7878	Ford	Fiesta	2001
JDM8776	Wolksvagen	Santana	2002
JJM3692	Chevrolet	Corsa	1999

SELECT Seleção

SELECT * FROM Taxi WHERE AnoFab > 2000

<u>Placa</u>	Marca	Modelo	AnoFab
D K L 4 5 9 8	W o l k s v a g e n	G o l	2 0 0 1
D K L 7 8 7 8	F o r d	F i e s t a	2 0 0 1
J D M 8 7 7 6	W o l k s v a g e n	S a n t a n a	2 0 0 2

Exercício 3

- Para as tabelas que você montou no exercício 1, escreva um comando SQL que retorne:
 - a) nomes de todas as mães
 - b) nomes de todas as mães com filhos maiores de 12 anos
- Esquema do Exercício 1
 - Pessoa(nome, nome_da_mãe, ano_nascimento, nome_cidade_natal)
 - nome_cidade_natal → CHE Cidade
 - Cidade(nome_cidade, sigla_estado)

SELECT LIKE

- SELECT ...
FROM <tabela₁>[,..., <tabela_n>]
WHERE <condição/junção>
LIKE <expressão_regular>
- Elementos da expressão regular
 - % → qualquer cadeia com 0 a n caracteres
 - _ → exatamente um caractere (qualquer)
 - = → caractere de escape
 - e.g., serve para encontrar um caractere _

LIKE

Questão

- Placas que comecem com DK

```
SELECT * FROM Taxi WHERE placa LIKE ?
```

- Placas com '7' na penultima posicao

```
SELECT * FROM Taxi WHERE placa LIKE ?
```

- Elementos da expressão regular

- % → qualquer cadeia com 0 a n caracteres
- _ → exatamente um caractere (qualquer)
- = → caractere de escape
 - e.g., serve para encontrar um caractere _

LIKE

Questão 1

- Placas que comecem com DK

```
SELECT * FROM Taxi WHERE placa LIKE 'DK%';
```

- Placas com '7' na penultima posicao

```
SELECT * FROM Taxi WHERE placa LIKE '%7_';
```

- Elementos da expressão regular

- % → qualquer cadeia com 0 a n caracteres
- _ → exatamente um caractere (qualquer)
- = → caractere de escape
 - e.g., serve para encontrar um caractere _

AS

alias (apelido)

- `SELECT <campo1> [AS] <alias1>`
`[,..., <campon> [AS] <aliasn>]`
...
- `SELECT ...`
`FROM <tabela1> [AS] <alias1>`
`[,..., <tabelan> [AS] <aliasn>]`
...

AS (alias / apelido) Tabelas

- Alias com o AS

```
SELECT Cl.CliId, Cl.Nome, Co.Placa, Co.DataPedido  
FROM Cliente AS Cl, Corrida AS Co  
WHERE Cl.CliId = Co.CliId;
```

- Alias sem o AS

```
SELECT Cl.CliId, Cl.Nome, Co.Placa, Co.DataPedido  
FROM Cliente Cl, Corrida Co  
WHERE Cl.CliId = Co.CliId;
```

AS (alias / apelido) Campos

- Alias com o AS

```
SELECT Cl.CliId AS id_cliente,  
       Cl.Nome AS nome_cliente, Co.Placa AS placa,  
       Co.DataPedido AS data_pedido  
FROM Cliente Cl, Corrida Co  
WHERE Cl.CliId = Co.CliId;
```

- Alias sem o AS

```
SELECT Cl.CliId id_cliente,  
       Cl.Nome nome_cliente, Co.Placa placa,  
       Co.DataPedido data_pedido  
FROM Cliente Cl, Corrida Co  
WHERE Cl.CliId = Co.CliId;
```

Questão 2

- Modelos de taxi existentes
- Qual a limitação desta solução?

```
SELECT Modelo  
FROM Taxi;
```

SELECT DISTINCT e ALL

- **SELECT DISTINCT ...**
 - não retorna tuplas repetidas
 - compatível com o modelo relacional

- **SELECT ALL ...**
 - cláusula implícita se não especificada
 - retorna todas as tuplas, mesmo se repetidas

DISTINCT

Questão 2

- Modelos de taxi existentes
- Sem DISTINCT

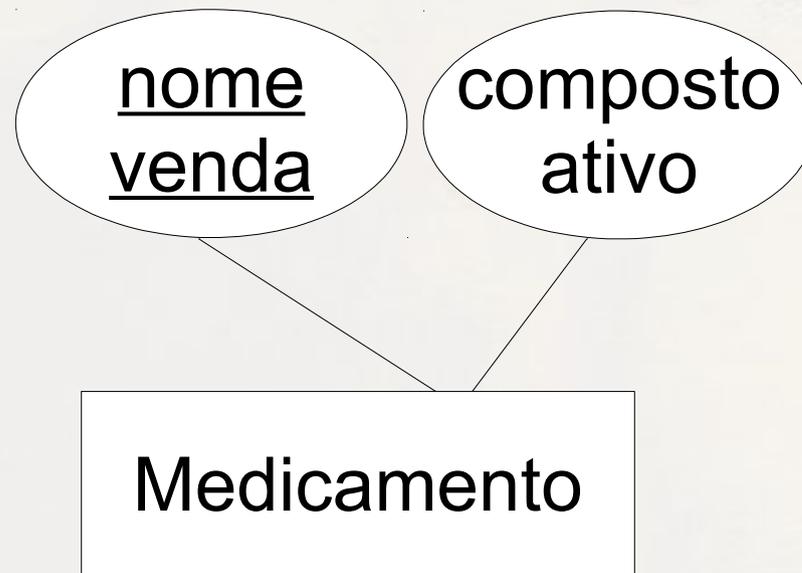
```
SELECT Modelo  
FROM Taxi;
```

- Com DISTINCT

```
SELECT DISTINCT Modelo  
FROM Taxi;
```

Exercício 4

- Liste todos os compostos ativos disponíveis
- Esquema:
 - medicamento (nomeVenda, compostoAtivo)



SELECT ORDER BY

- **SELECT ...
ORDER BY <campo₁> [, ..., <campo_n>]**
- **Nome dos clientes em ordem crescente**

```
SELECT Nome  
FROM Cliente  
ORDER BY Nome;
```

Produto Cartesiano

- **SELECT ...**
FROM <tabela₁>, <tabela₂>
<não há condição que ligue tabelas>
- Não há associação de atributo da <tabela₁>
com atributo da <tabela₂>

Produto Cartesiano

```
SELECT Cliente.CliId, Cliente.Nome,  
Corrida.CliId, Corrida.Placa,  
Corrida.DataPedido  
FROM Cliente, Corrida
```

<u>C liId</u>	<u>N o m e</u>
1 5 3 2	A s d r ú b a l
1 7 5 5	D o r i a n a
1 7 8 0	Q u i n c a s

<u>C liId</u>	<u>P l a c a</u>	<u>D a t a P e d i d o</u>
1 7 5 5	D A E 6 5 3 4	1 5 / 0 2 / 2 0 0 3
1 9 8 2	J D M 8 7 7 6	1 8 / 0 2 / 2 0 0 3

Produto Cartesiano

```
SELECT Cliente.CliId, Cliente.Nome,  
Corrida.CliId, Corrida.Placa,  
Corrida.DataPedido  
FROM Cliente, Corrida
```

<u>C liId</u>	N o m e
1 5 3 2	A s d r ú b a l
1 7 5 5	D o r i a n a
1 7 8 0	Q u i n c a s

<u>C lId</u>	<u>P l a c a</u>	<u>D a t a P e d i d o</u>
1 7 5 5	D A E 6 5 3 4	1 5 / 0 2 / 2 0 0 3
1 9 8 2	J D M 8 7 7 6	1 8 / 0 2 / 2 0 0 3

(C l i d)	N o m e	(C l i d)	P l a c a	D a t a P e d i d o
1 5 3 2	A s d r ú b a l	1 7 5 5	D A E 6 5 3 4	1 5 / 0 2 / 2 0 0 3
1 5 3 2	A s d r ú b a l	1 9 8 2	J D M 8 7 7 6	1 8 / 0 2 / 2 0 0 3

Produto Cartesiano

```
SELECT Cliente.CliId, Cliente.Nome,
       Corrida.CliId, Corrida.Placa,
       Corrida.DataPedido
FROM Cliente, Corrida
```

<u>C liId</u>	N o m e
1 5 3 2	A s d r ú b a l
1 7 5 5	D o r i a n a
1 7 8 0	Q u i n c a s

<u>C lId</u>	<u>P l a c a</u>	<u>D a t a P e d i d o</u>
1 7 5 5	D A E 6 5 3 4	1 5 / 0 2 / 2 0 0 3
1 9 8 2	J D M 8 7 7 6	1 8 / 0 2 / 2 0 0 3

(C l i d)	N o m e	(C l i d)	P l a c a	D a t a P e d i d o
1 5 3 2	A s d r ú b a l	1 7 5 5	D A E 6 5 3 4	1 5 / 0 2 / 2 0 0 3
1 5 3 2	A s d r ú b a l	1 9 8 2	J D M 8 7 7 6	1 8 / 0 2 / 2 0 0 3
1 7 5 5	D o r i a n a	1 7 5 5	D A E 6 5 3 4	1 5 / 0 2 / 2 0 0 3
1 7 5 5	D o r i a n a	1 9 8 2	J D M 8 7 7 6	1 8 / 0 2 / 2 0 0 3

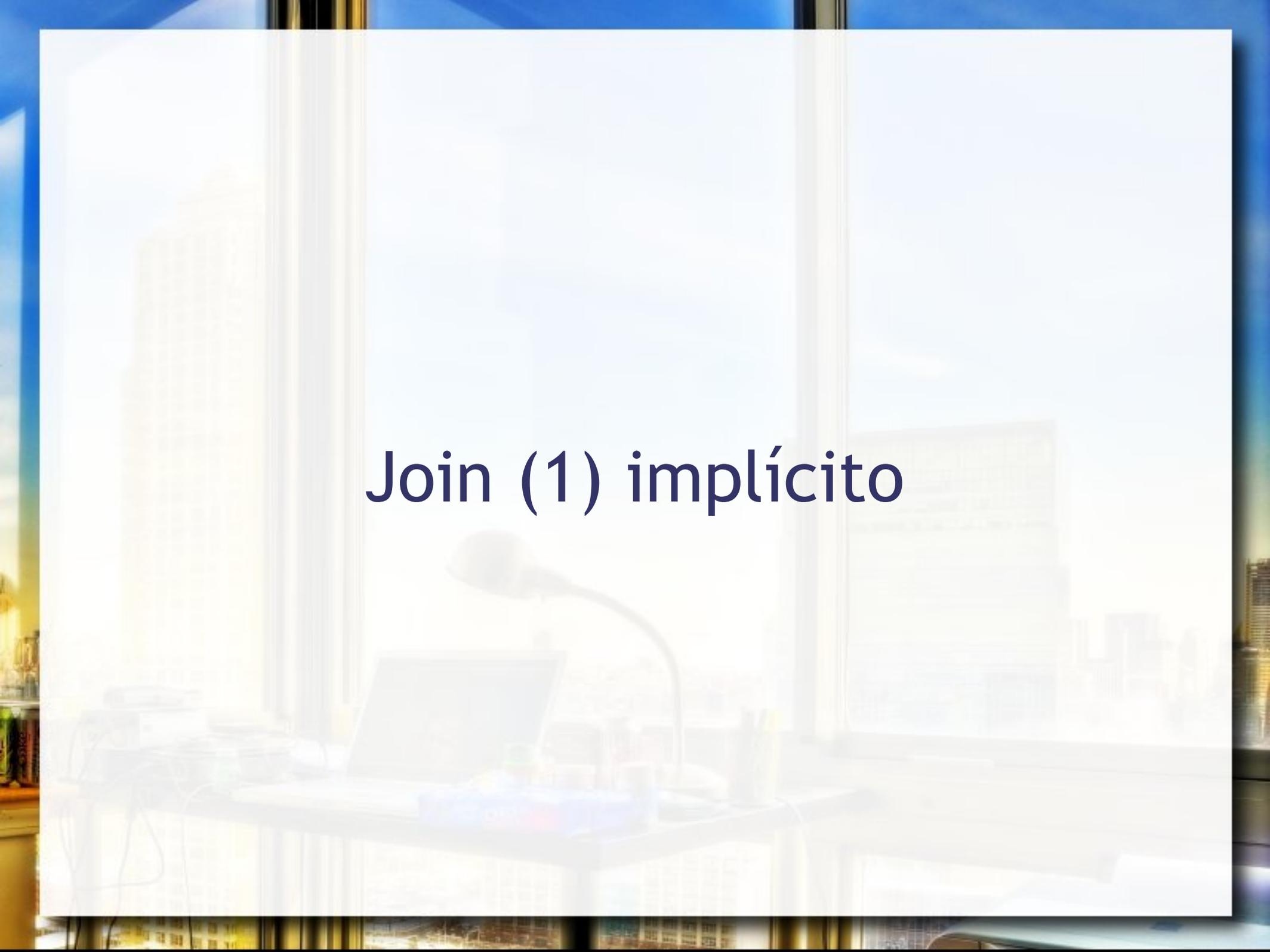
Produto Cartesiano

```
SELECT Cliente.CliId, Cliente.Nome,
       Corrida.CliId, Corrida.Placa,
       Corrida.DataPedido
FROM Cliente, Corrida
```

<u>C liId</u>	N o m e
1 5 3 2	A s d r ú b a l
1 7 5 5	D o r i a n a
1 7 8 0	Q u i n c a s

<u>C lId</u>	<u>P l a c a</u>	<u>D a t a P e d i d o</u>
1 7 5 5	D A E 6 5 3 4	1 5 / 0 2 / 2 0 0 3
1 9 8 2	J D M 8 7 7 6	1 8 / 0 2 / 2 0 0 3

(C liId)	N o m e	(C lId)	P l a c a	D a t a P e d i d o
1 5 3 2	A s d r ú b a l	1 7 5 5	D A E 6 5 3 4	1 5 / 0 2 / 2 0 0 3
1 5 3 2	A s d r ú b a l	1 9 8 2	J D M 8 7 7 6	1 8 / 0 2 / 2 0 0 3
1 7 5 5	D o r i a n a	1 7 5 5	D A E 6 5 3 4	1 5 / 0 2 / 2 0 0 3
1 7 5 5	D o r i a n a	1 9 8 2	J D M 8 7 7 6	1 8 / 0 2 / 2 0 0 3
1 7 8 0	Q u i n c a s	1 7 5 5	D A E 6 5 3 4	1 5 / 0 2 / 2 0 0 3
1 7 8 0	Q u i n c a s	1 9 8 2	J D M 8 7 7 6	1 8 / 0 2 / 2 0 0 3



Join (1) implícito

Junção - Join (1)

- **SELECT ...**
FROM <tabela₁>, <tabela₂>
WHERE <tabela₁>.<atr> = <tabela₂>.<atr>
- **Join implícito**

Join (1)

```
SELECT Cliente.CliId, Cliente.Nome,  
Corrida.CliId, Corrida.Placa,  
Corrida.DataPedido  
FROM Cliente, Corrida  
WHERE Cliente.CliId = Corrida.CliId
```

(C liId)	N o m e	(C liId)	P l a c a	D a t a P e d i d o
1 5 3 2	A s d r ú b a l	1 7 5 5	D A E 6 5 3 4	1 5 / 0 2 / 2 0 0 3
1 5 3 2	A s d r ú b a l	1 9 8 2	J D M 8 7 7 6	1 8 / 0 2 / 2 0 0 3
1 7 5 5	D o r i a n a	1 7 5 5	D A E 6 5 3 4	1 5 / 0 2 / 2 0 0 3
1 7 5 5	D o r i a n a	1 9 8 2	J D M 8 7 7 6	1 8 / 0 2 / 2 0 0 3
1 7 8 0	Q u i n c a s	1 7 5 5	D A E 6 5 3 4	1 5 / 0 2 / 2 0 0 3
1 7 8 0	Q u i n c a s	1 9 8 2	J D M 8 7 7 6	1 8 / 0 2 / 2 0 0 3

Join (1)

```
SELECT Cliente.CliId, Cliente.Nome,  
Corrida.CliId, Corrida.Placa,  
Corrida.DataPedido  
FROM Cliente, Corrida  
WHERE Cliente.CliId = Corrida.CliId
```

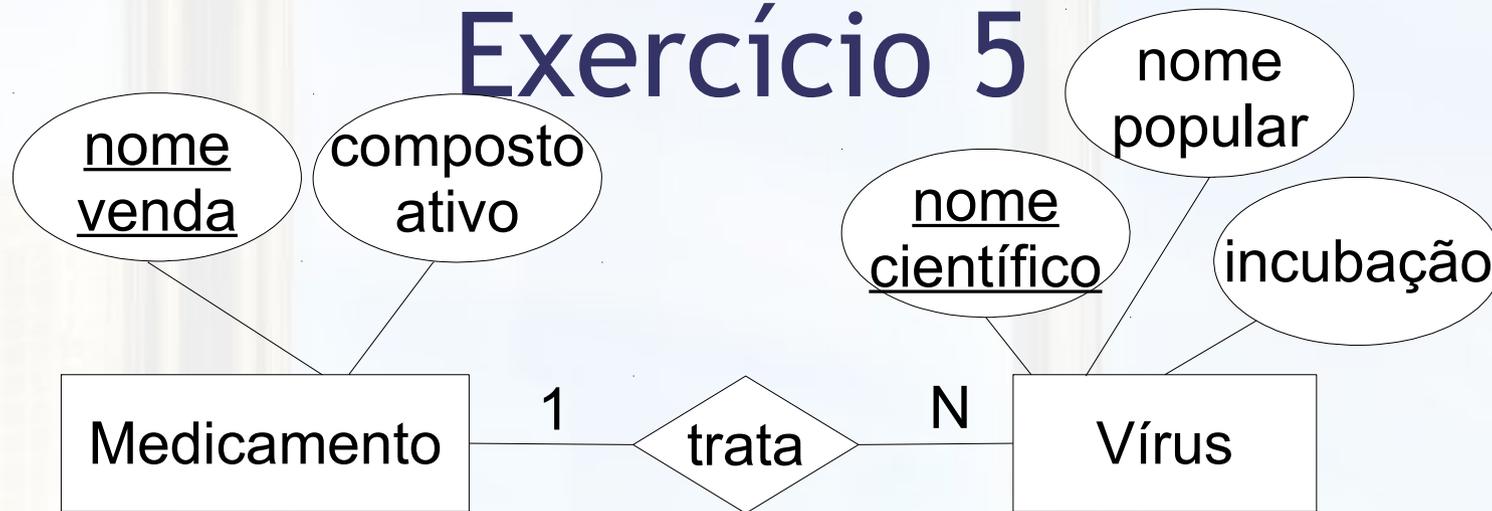
(C liId)	N o m e	(C liId)	P l a c a	D a t a P e d i d o
1 5 3 2	A s d r ú b a l	1 7 5 5	D A E 6 5 3 4	1 5 / 0 2 / 2 0 0 3
1 5 3 2	A s d r ú b a l	1 9 8 2	J D M 8 7 7 6	1 8 / 0 2 / 2 0 0 3
1 7 5 5	D o r i a n a	1 7 5 5	D A E 6 5 3 4	1 5 / 0 2 / 2 0 0 3
1 7 5 5	D o r i a n a	1 9 8 2	J D M 8 7 7 6	1 8 / 0 2 / 2 0 0 3
1 7 8 0	Q u i n c a s	1 7 5 5	D A E 6 5 3 4	1 5 / 0 2 / 2 0 0 3
1 7 8 0	Q u i n c a s	1 9 8 2	J D M 8 7 7 6	1 8 / 0 2 / 2 0 0 3

Join (1)

```
SELECT Cliente.CliId, Cliente.Nome,  
Corrida.CliId, Corrida.Placa,  
Corrida.DataPedido  
FROM Cliente, Corrida  
WHERE Cliente.CliId = Corrida.CliId
```

(C liId)	N o m e	(C liId)	P l a c a	D a t a P e d i d o
1755	D o r i a n a	1755	D A E 6 5 3 4	15/02/2003

Exercício 5



medicamento (nomeVenda, compostoAtivo)

virus (nomeCientifico, nomePopular, incubacao, **nomeVendaMedicamento**)

- CHE: nomeVendaMedicamento para medicamento

- Para as tabelas acima escreva uma sentença SQL que retorne o nome popular dos vírus:
 - a) tratados pelo medicamento de nome de venda W
 - b) tratados pelo medicamento de composto ativo X

Exercício Casa 1

- Para as tabelas que você montou no exercício 1, escreva um comando SQL que retorne:
 - nomes de parentes que nasceram no mesmo estado que você
 - retorne todos os primos por parte de mãe, que você for capaz de inferir a partir da tabela

Questão 3

- Qual o modelo de Taxi para cada Corrida?

Cliente (C)

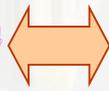
<u>C liId</u>	N o m e	C P F
1532	A sdrúb al	448.754.253-65
1755	D oriana	567.387.387-44
1780	Q uincas	546.373.762-02



Táxi (TX)



<u>P l a c a</u>	M a r c a	M o d e l o	A n o F a b
D A E 6534	F o r d	F i e s t a	1999
D K L 4598	W o l k s v a g e n	G o l	2001
D K L 7878	F o r d	F i e s t a	2001
J D M 8776	W o l k s v a g e n	S a n t a n a	2002
J J M 3692	C h e v r o l e t	C o r s a	1999



Corrida (R1)

<u>C liId</u>	<u>P l a c a</u>	<u>D a t a P e d i d o</u>
1755	D A E 6534	15/02/2003
1982	J D M 8776	18/02/2003

Modelo de Taxi para cada Corrida

```

SELECT Co.DataPedido, Co.Placa, T.Modelo
FROM Corrida Co, Taxi T
WHERE Co.Placa = T.Placa;
    
```

Cliente (C)

<u>C liId</u>	N ome	C P F
1532	A sdrúb al	448.754.253-65
1755	D oriana	567.387.387-44
1780	Q uincas	546.373.762-02



Táxi (TX)



<u>P laca</u>	M arca	M odelo	A noFab
D A E 6534	Ford	Fiesta	1999
D K L 4598	W olkswagen	G ol	2001
D K L 7878	Ford	Fiesta	2001
J D M 8776	W olkswagen	S antana	2002
J J M 3692	C hevrolet	C orsa	1999



Corrida (R1)

<u>C liId</u>	<u>P laca</u>	<u>D ataPedido</u>
1755	D A E 6534	15/02/2003
1982	J D M 8776	18/02/2003

Desafio

- Quais os modelos de Taxi tomados por cada Cliente?

Cliente (C)

<u>C liId</u>	N o m e	C P F
1532	A sdrúb al	448.754.253-65
1755	D oriana	567.387.387-44
1780	Q uincas	546.373.762-02



Táxi (TX)



<u>P l a c a</u>	M a r c a	M o d e l o	A n o F a b
D A E 6534	F o r d	F i e s t a	1999
D K L 4598	W o l k s v a g e n	G o l	2001
D K L 7878	F o r d	F i e s t a	2001
J D M 8776	W o l k s v a g e n	S a n t a n a	2002
J J M 3692	C h e v r o l e t	C o r s a	1999



Corrida (R1)

<u>C liId</u>	<u>P l a c a</u>	<u>D a t a P e d i d o</u>
1755	D A E 6534	15/02/2003
1982	J D M 8776	18/02/2003

Modelos de Taxi por Cliente (estagio 1)

```

SELECT Cl.Nome, Co.DataPedido, Co.Placa, T.Modelo
FROM Cliente Cl, Corrida Co, Taxi T
WHERE Cl.CliId = Co.CliId AND Co.Placa = T.Placa;
    
```

Cliente (C)

<u>C liId</u>	N o m e	C P F
1532	A sdrúb al	448.754.253-65
1755	D oriana	567.387.387-44
1780	Q uincas	546.373.762-02



Táxi (TX)



<u>P l a c a</u>	M a r c a	M o d e l o	A n o F a b
D A E 6534	Ford	Fiesta	1999
D K L 4598	W o l k s v a g e n	G o l	2001
D K L 7878	Ford	Fiesta	2001
J D M 8776	W o l k s v a g e n	S a n t a n a	2002
J J M 3692	C h e v r o l e t	C o r s a	1999



Corrida (R1)

<u>C l i d</u>	<u>P l a c a</u>	<u>D a t a P e d i d o</u>
1755	D A E 6534	15/02/2003
1982	J D M 8776	18/02/2003

Modelos de Taxi por Cliente (estagio 2)

```
SELECT DISTINCT Cl.Nome, T.Modelo
FROM Cliente Cl, Corrida Co, Taxi T
WHERE Cl.CliId = Co.CliId AND Co.Placa = T.Placa;
```

Cliente (C)

<u>C liId</u>	N o m e	C P F
1532	A sdrúb al	448.754.253-65
1755	D oriana	567.387.387-44
1780	Q uincas	546.373.762-02



Táxi (TX)



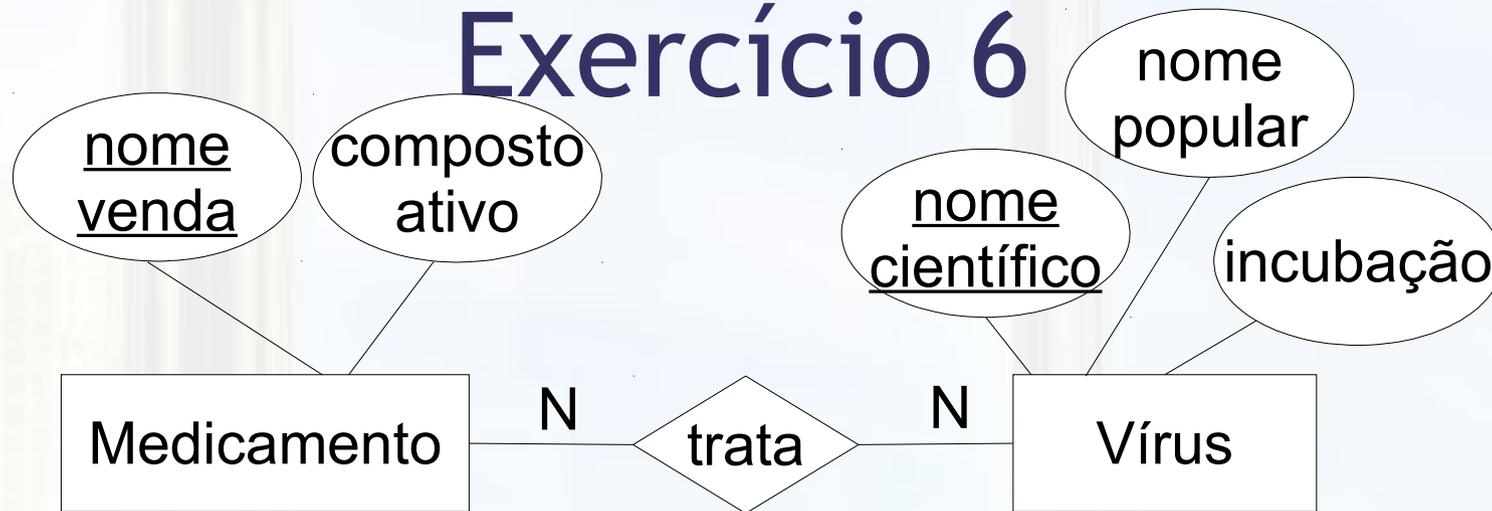
<u>P l a c a</u>	M a r c a	M o d e l o	A n o F a b
D A E 6534	Ford	Fiesta	1999
D K L 4598	Wolkswagen	Gol	2001
D K L 7878	Ford	Fiesta	2001
J D M 8776	Wolkswagen	Santana	2002
J J M 3692	Chevrolet	Corsa	1999



Corrida (R1)

<u>C liId</u>	<u>P l a c a</u>	<u>D a t a P e d i d o</u>
1755	D A E 6534	15/02/2003
1982	J D M 8776	18/02/2003

Exercício 6



medicamento (nomeVenda, compostoAtivo)

virus (nomeCientifico, nomePopular, incubacao)

trata (nomeVendaMedicamento, nomeCientificoVirus)

- CHE: nomeVendaMedicamento para medicamento
- CHE: nomeCientificoVirus para virus
- Para as tabelas acima escreva sentença SQL que retorne:
 - nome popular dos vírus tratados pelo medicamento de composto ativo X

Questão 3a

ORDER BY

- Modelos de taxi e os clientes que os tomaram
- ?

```
SELECT DISTINCT Cl.Nome, T.Modelo
FROM Cliente Cl, Corrida Co, Taxi T
WHERE Cl.CliId = Co.CliId AND
      Co.Placa = T.Placa
ORDER BY Cl.Modelo;
```

- ?

```
SELECT DISTINCT Cl.Nome, T.Modelo
FROM Cliente Cl, Corrida Co, Taxi T
WHERE Cl.CliId = Co.CliId AND
      Co.Placa = T.Placa
ORDER BY T.Modelo, Cl.Nome;
```

Questão 3a

ORDER BY

- Modelos de taxi e os clientes que os tomaram
- Ordena por Modelo, mas não por Cliente

```
SELECT DISTINCT Cl.Nome, T.Modelo
FROM Cliente Cl, Corrida Co, Taxi T
WHERE Cl.CliId = Co.CliId AND
      Co.Placa = T.Placa
ORDER BY Cl.Modelo;
```

- Ordena por Modelo e, para cada Modelo, por Cliente

```
SELECT DISTINCT Cl.Nome, T.Modelo
FROM Cliente Cl, Corrida Co, Taxi T
WHERE Cl.CliId = Co.CliId AND
      Co.Placa = T.Placa
ORDER BY T.Modelo, Cl.Nome;
```

Questão 3b

ORDER BY

- Cliente e os modelos de taxi tomados?

Questão 3b

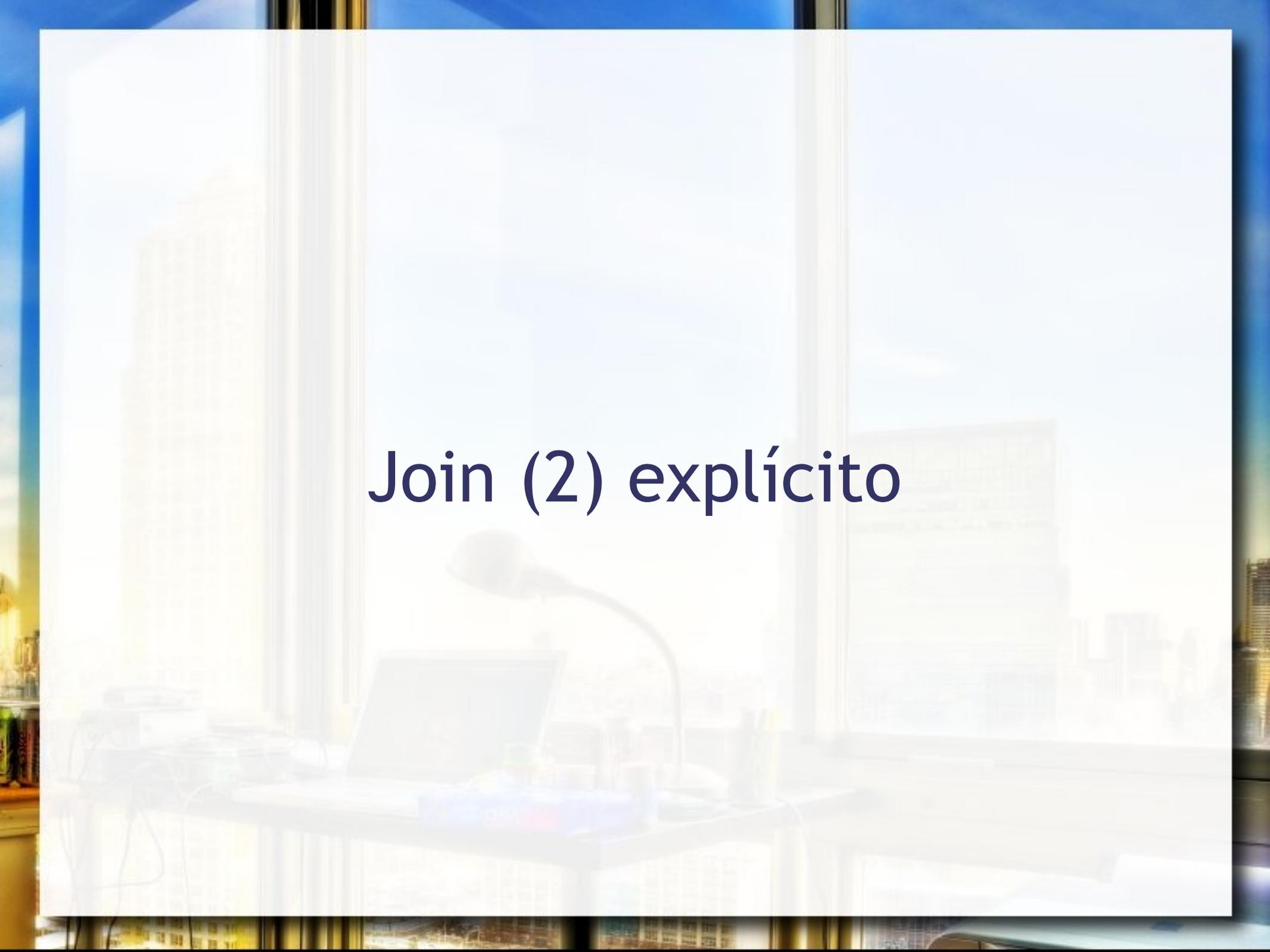
ORDER BY

- Cliente e os modelos de taxi tomados
- Ordena por Cliente, mas não por Modelo

```
SELECT DISTINCT Cl.Nome, T.Modelo
FROM Cliente Cl, Corrida Co, Taxi T
WHERE Cl.CliId = Co.CliId AND
      Co.Placa = T.Placa
ORDER BY Cl.Nome;
```

- Ordena por Cliente e, para cada Cliente, por Modelo

```
SELECT DISTINCT Cl.Nome, T.Modelo
FROM Cliente Cl, Corrida Co, Taxi T
WHERE Cl.CliId = Co.CliId AND
      Co.Placa = T.Placa
ORDER BY Cl.Nome, T.Modelo;
```



Join (2) explícito

Join (2)

- **SELECT ...**
 FROM <tabela> JOIN <tabela>
 ON <condição> ...
- **Tipo clássico de join explicitado**
- **Também conhecido como INNER JOIN**

Natural Join

- **SELECT ...**
FROM <tabela> NATURAL JOIN <tabela>
- **Condição não especificada**
- **EQUIJOIN:** Verifica igualdade de cada par de atributos com o mesmo nome

Outer Join

- **SELECT ...**
FROM <tabela> <join> <tabela>
ON <condição> ...
- <join>
 - LEFT JOIN - toda tupla à esquerda aparece
 - RIGHT JOIN - toda tupla à direita aparece
 - FULL JOIN - toda tupla aparece

DELETE

- DELETE FROM <tabela₁>
WHERE <condição>
- executeUpdate(...)

Excluir as Corridas feitas pelo Taxi DAE6534

DELETE FROM Corrida Co
WHERE Co.Placa = 'DAE6534';

Cliente (C)

<u>C liId</u>	N o m e	C P F
1532	A sdrúbal	448.754.253-65
1755	D oriana	567.387.387-44
1780	Q uincas	546.373.762-02



Táxi (TX)



<u>P l a c a</u>	M a r c a	M o d e l o	A n o F a b
D A E 6 5 3 4	Ford	Fiesta	1999
D K L 4 5 9 8	W o l k s v a g e n	G o l	2001
D K L 7 8 7 8	Ford	Fiesta	2001
J D M 8 7 7 6	W o l k s v a g e n	S a n t a n a	2002
J J M 3 6 9 2	C h e v r o l e t	C o r s a	1999



Corrida (R1)

<u>C liId</u>	<u>P l a c a</u>	<u>D a t a P e d i d o</u>
1755	D A E 6 5 3 4	15/02/2003
1982	J D M 8 7 7 6	18/02/2003

UPDATE

- UPDATE <tabela>
SET <campo₁>=<valor₁>
[,..., <campo_n>=<valor_n>]
WHERE <condição>
- executeUpdate(...)

Mudar o Ano de Fabricação do Táxi DKL4598 para 2002

UPDATE Taxi T
SET T.AnoFab = 2002
WHERE T.Placa = 'DKL4598';

Cliente (C)

<u>C liId</u>	N ome	C P F
1532	A sdrúb al	448.754.253-65
1755	D oriana	567.387.387-44
1780	Q uincas	546.373.762-02



Táxi (TX)

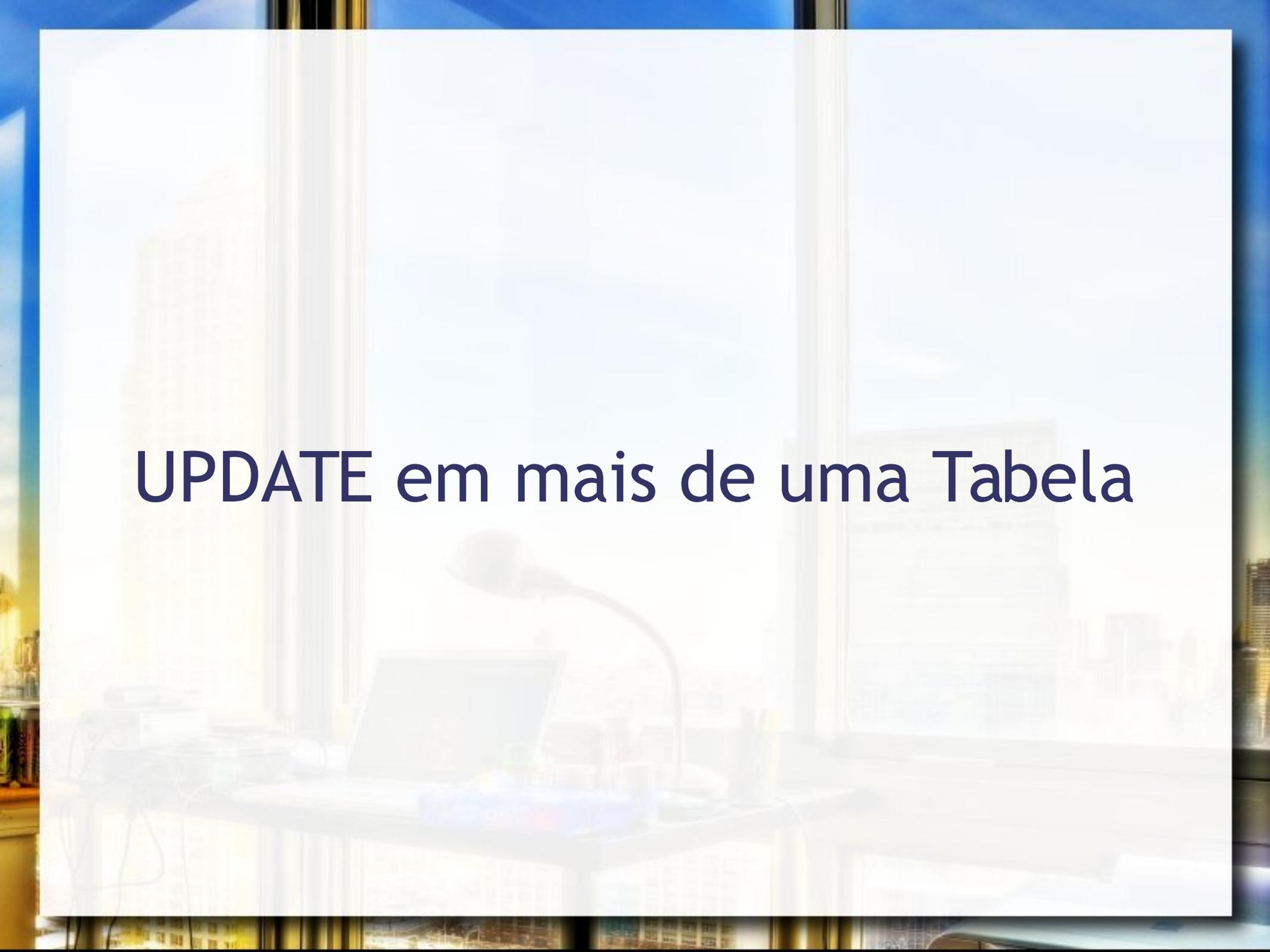


<u>P laca</u>	M arca	M odelo	A noFab
D A E 6534	Ford	Fiesta	1999
D K L4598	W olkswagen	G ol	2001
D K L7878	Ford	Fiesta	2001
J D M 8776	W olkswagen	S antana	2002
J J M 3692	C hevrolet	C orsa	1999



Corrida (R1)

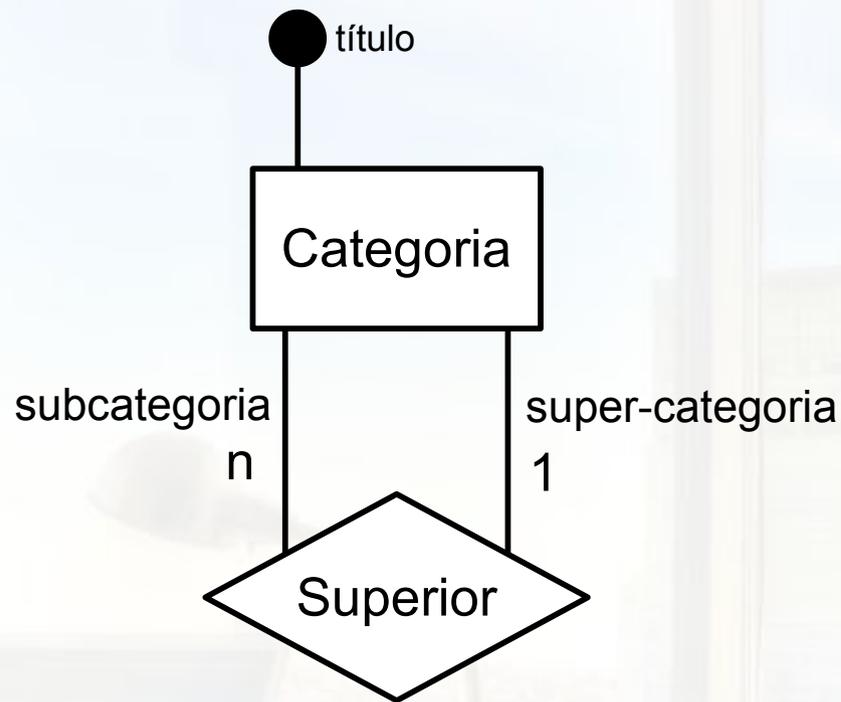
<u>C liId</u>	<u>P laca</u>	<u>D ataPedido</u>
1755	D A E 6534	15/02/2003
1982	J D M 8776	18/02/2003



UPDATE em mais de uma Tabela

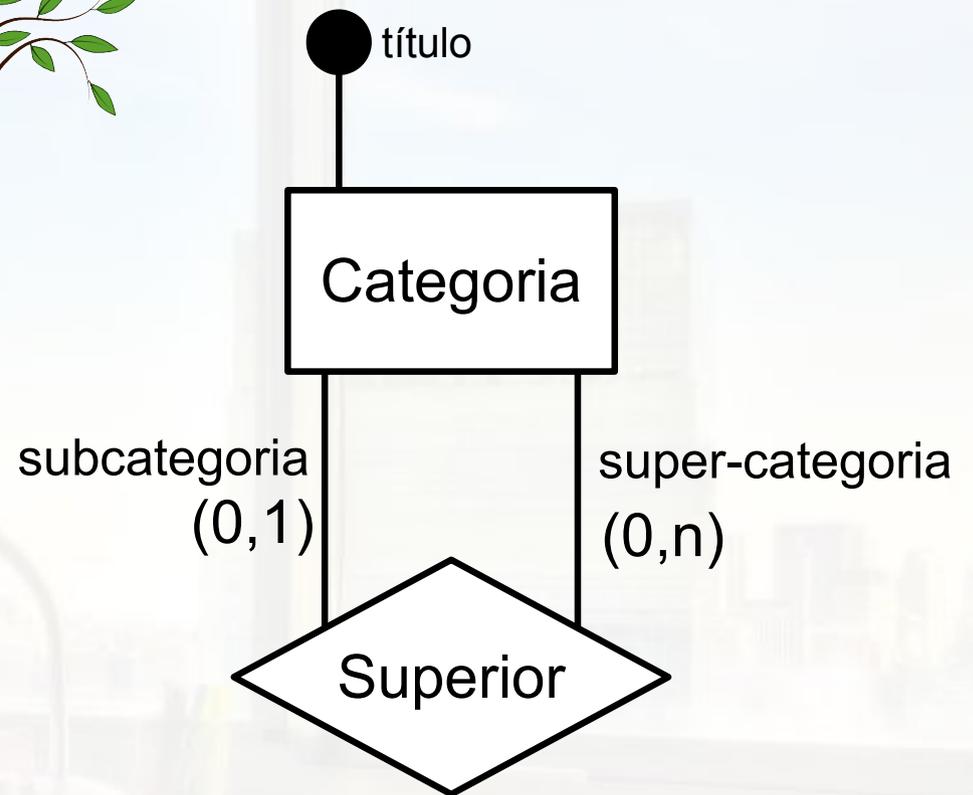
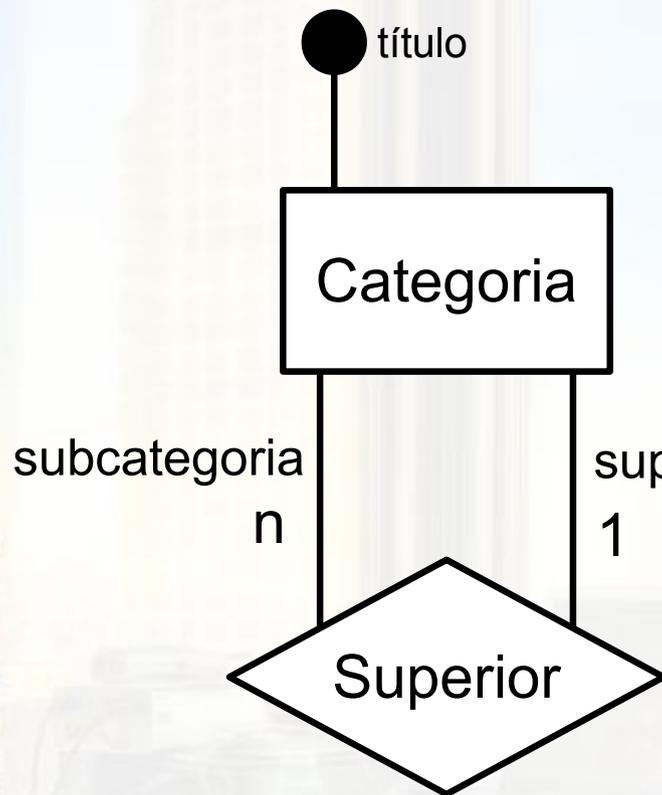
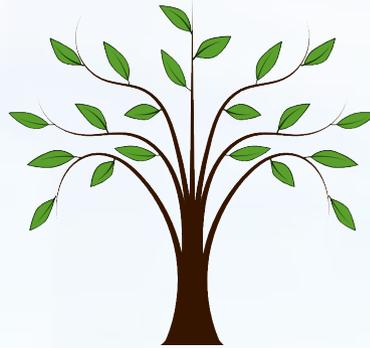
Categorias de Marcadores

Modelo ER



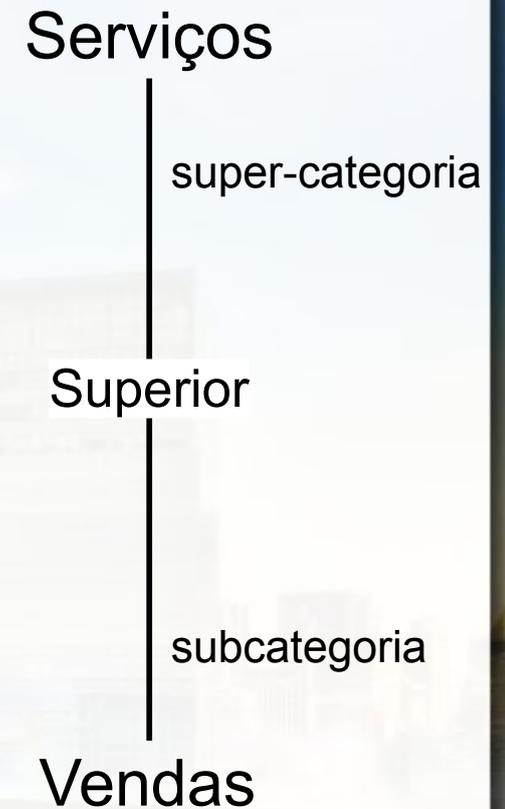
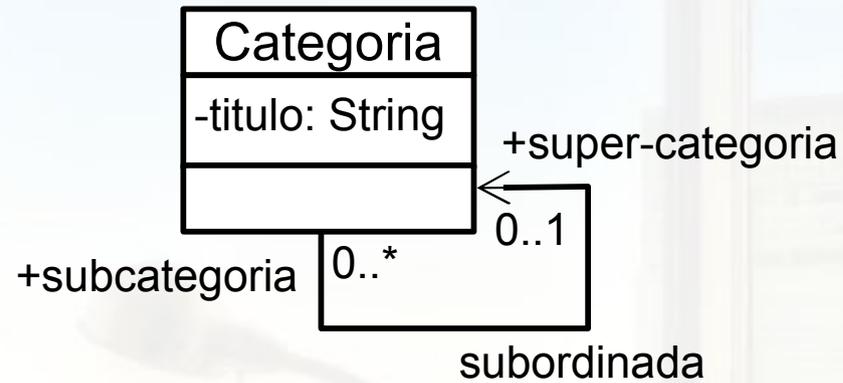
Categorías de Marcadores

Modelo ER



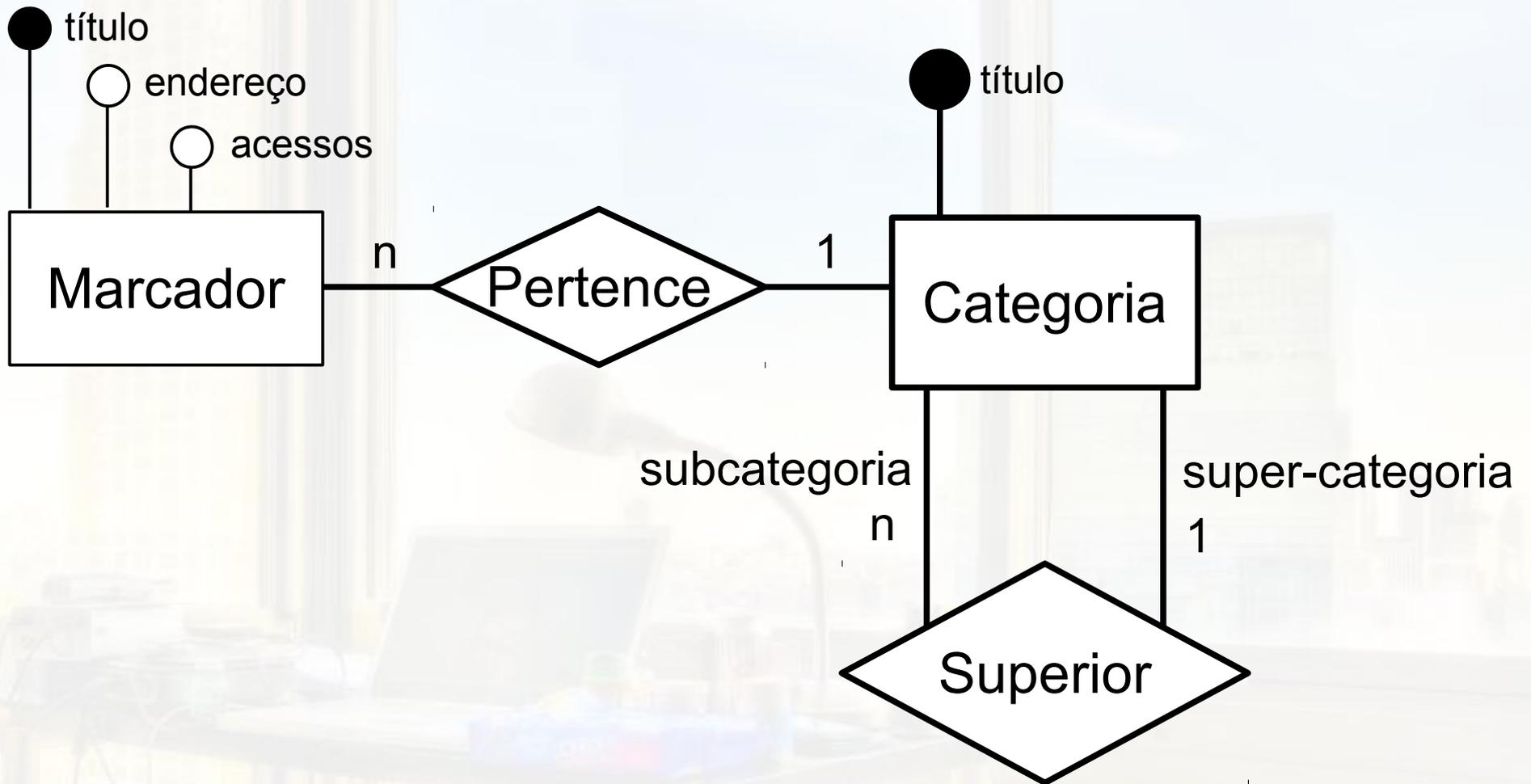
Categorias de Marcadores

Modelo UML



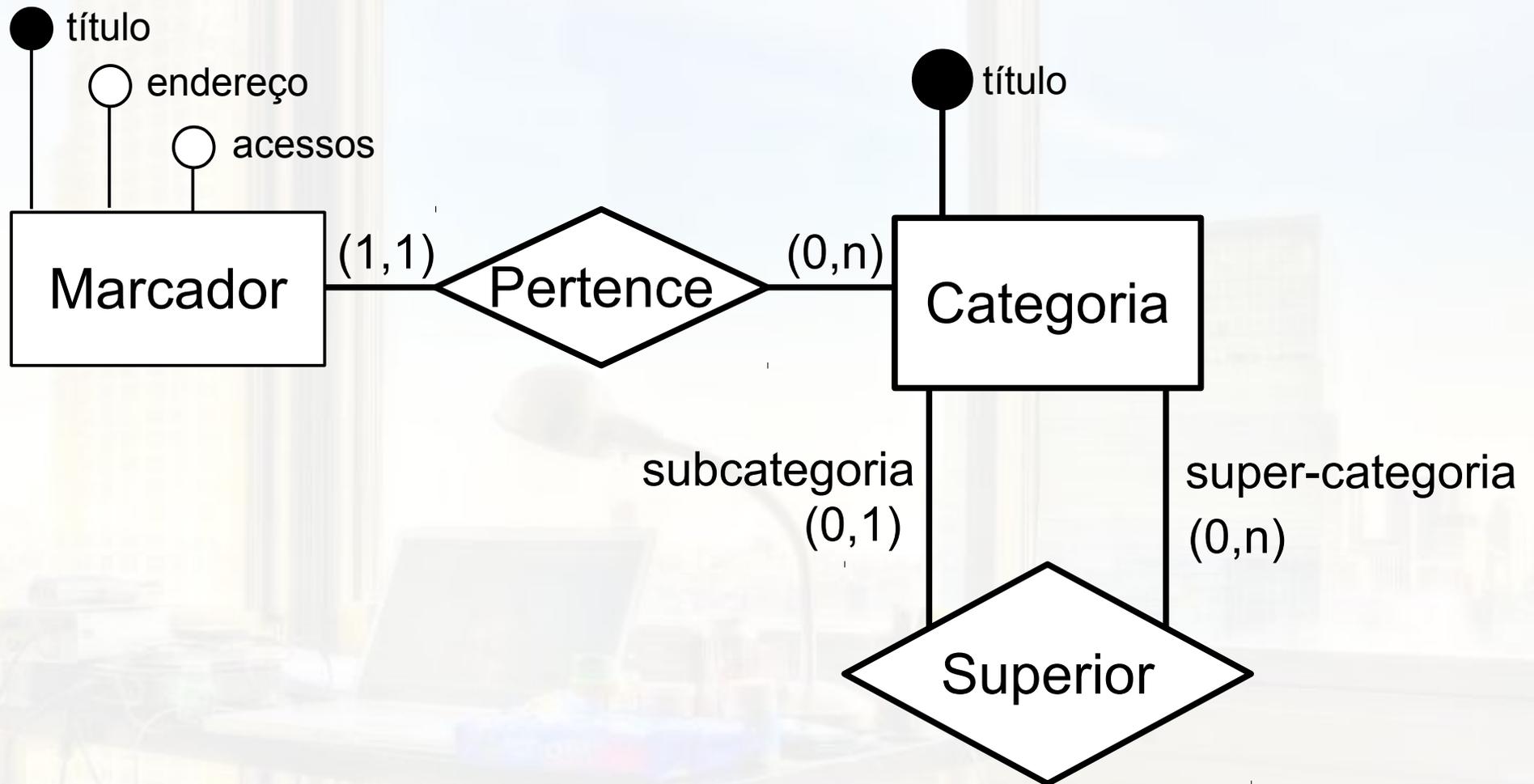
Marcadores e Categorias

Modelo ER



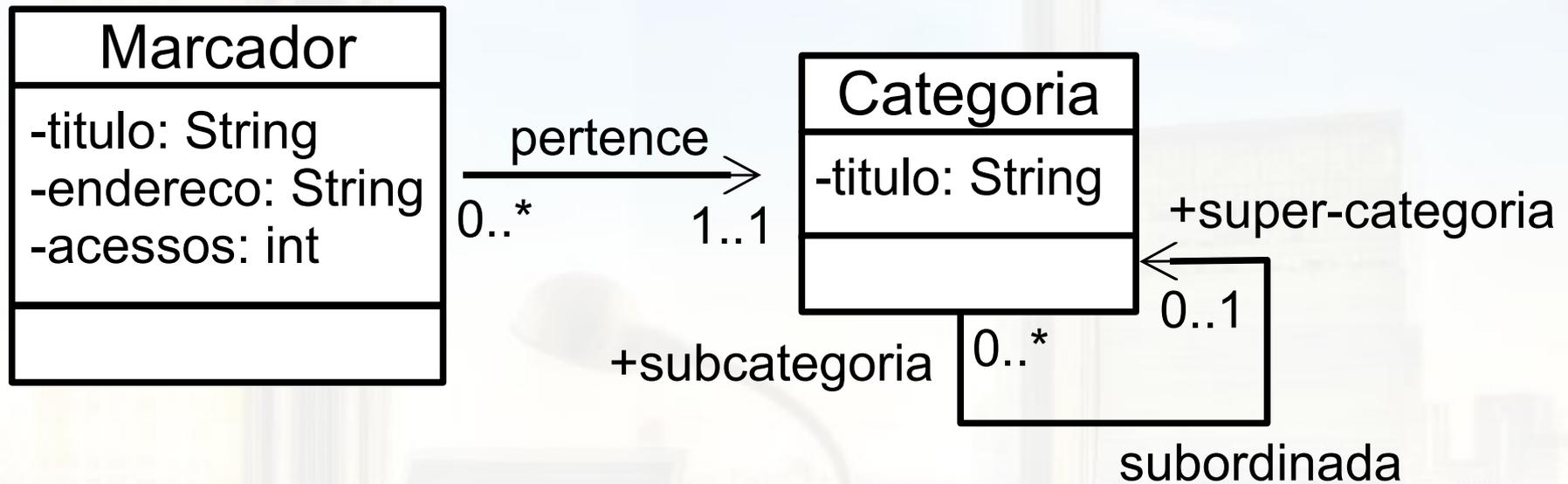
Marcadores e Categorias

Modelo ER



Marcadores e Categorias

Modelo UML



Marcadores e Categorias

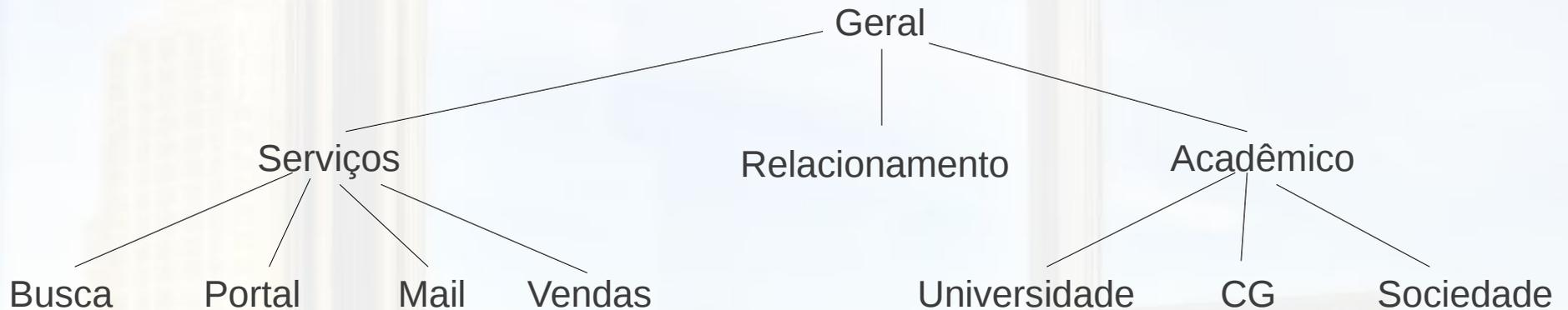
Modelo Relacional

Marcador (Titulo, Endereco, Acessos, Categoria)

Titulo	Endereco	Acessos	Categoria
Terra	http://www.terra.com.br	295	Portal
POVRay	http://www.povray.org	2	CG
SBC	http://www.sbc.org.br	26	Sociedade
Correios	http://www.correios.com.br	45	Serviços
GMail	http://www.gmail.com	296	Mail
Google	http://www.google.com	1590	Busca
Yahoo	http://www.yahoo.com	134	Serviços
Orkut	http://www.orkut.com	45	Serviços
iBahia	http://www.ibahia.com	3	Portal
Submarino	http://www.submarino.com.br	320	Serviços

Tabela Taxonomia

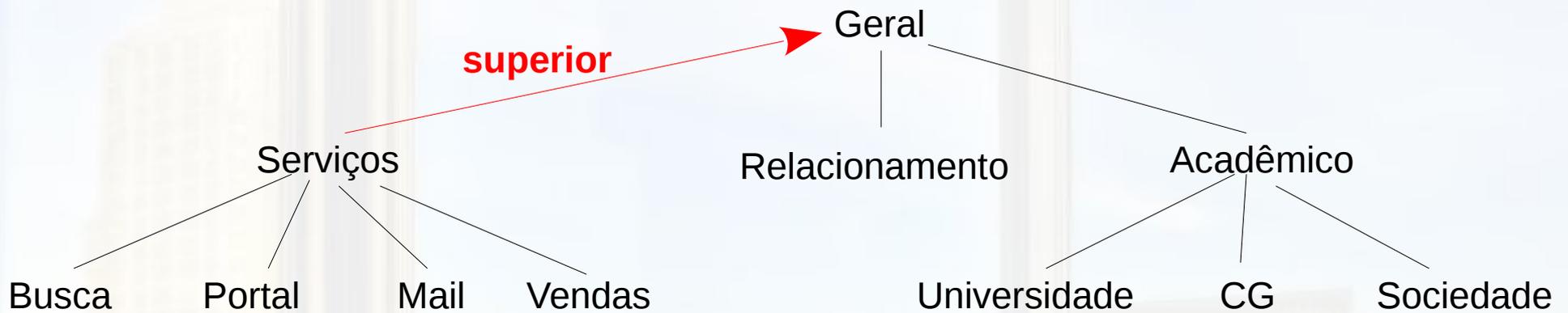
Modelo Relacional



Categoria	Superior
Geral	
Serviços	Geral
Acadêmico	Geral
Relacionamento	Geral
Busca	Serviços
Portal	Serviços
Mail	Serviços
Vendas	Serviços
Universidade	Acadêmico
CG	Acadêmico
Sociedade	Acadêmico

Tabela Taxonomia

Modelo Relacional



Categoria	Superior
Geral	
Serviços	Geral
Acadêmico	Geral
Relacionamento	Geral
Busca	Serviços
Portal	Serviços
Mail	Serviços
Vendas	Serviços
Universidade	Acadêmico
CG	Acadêmico
Sociedade	Acadêmico

Marcadores e Categorias

Modelo Relacional

Marcador (Titulo, Acessos, Endereco, Categoria)

- Categoria: chave estrangeira para Taxonomia

Taxonomia (Categoria, Superior)

Marcadores e Categorias

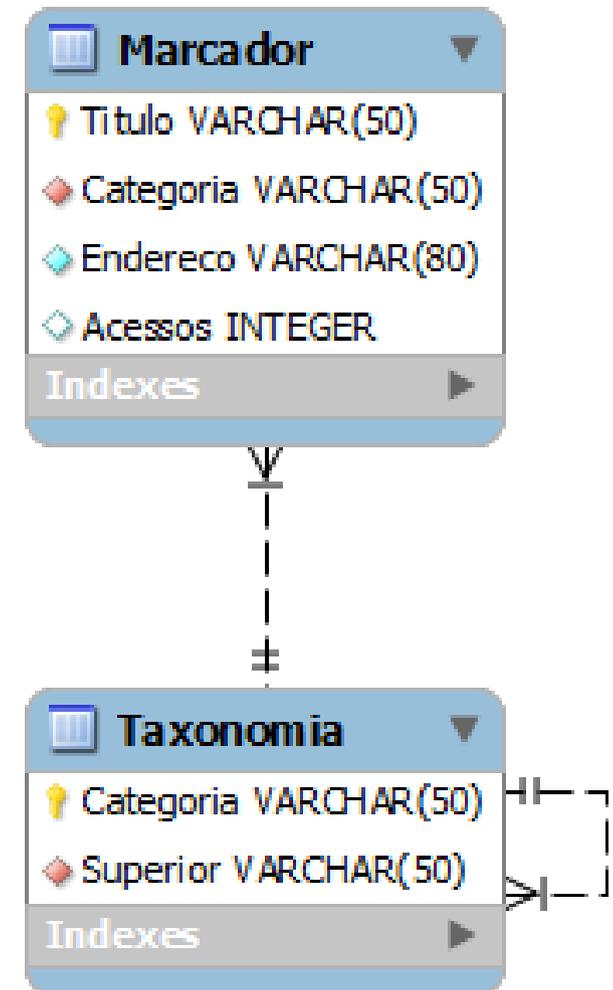
Diagrama Relacional (notação pé de galinha)

Marcador (Titulo, Acessos, Endereco, Categoria)

- Categoria: chave estrangeira para Taxonomia

Taxonomia (Categoria, Superior)

- Superior: chave estrangeira para Taxonomia



Questão 4

SQL

- UPDATE Marcadores
SET Categoria = <nova>
WHERE Categoria = <antiga>
- UPDATE Taxonomia
SET Categoria = <nova>
WHERE Categoria = <antiga>
- UPDATE Taxonomia
SET Superior = <nova>
WHERE Superior = <antiga>

Exercício para Casa 2

- Retomando os seguintes esquemas:
 - Pessoa(nome, nome_da_mãe, ano_nascimento, nome_cidade_natal)
 - nome_cidade_natal → CHE Cidade
 - Cidade(nome_cidade, sigla_estado)
- É possível especificar um comando SQL de criação da tabela Pessoa que permita mudar o nome de uma cidade nas tabelas Pessoa e Cidade com um único comando SQL?
- Se sim, escreva o(s) comando(s) CREATE necessários para isso e a sentença SQL de mudança do nome da cidade.



Prepared Statement

Utilizando o PreparedStatement

- `SELECT FROM Marcadores
WHERE Titulo = ?`
- `<comando>.setString(<numero>, <valor>)`

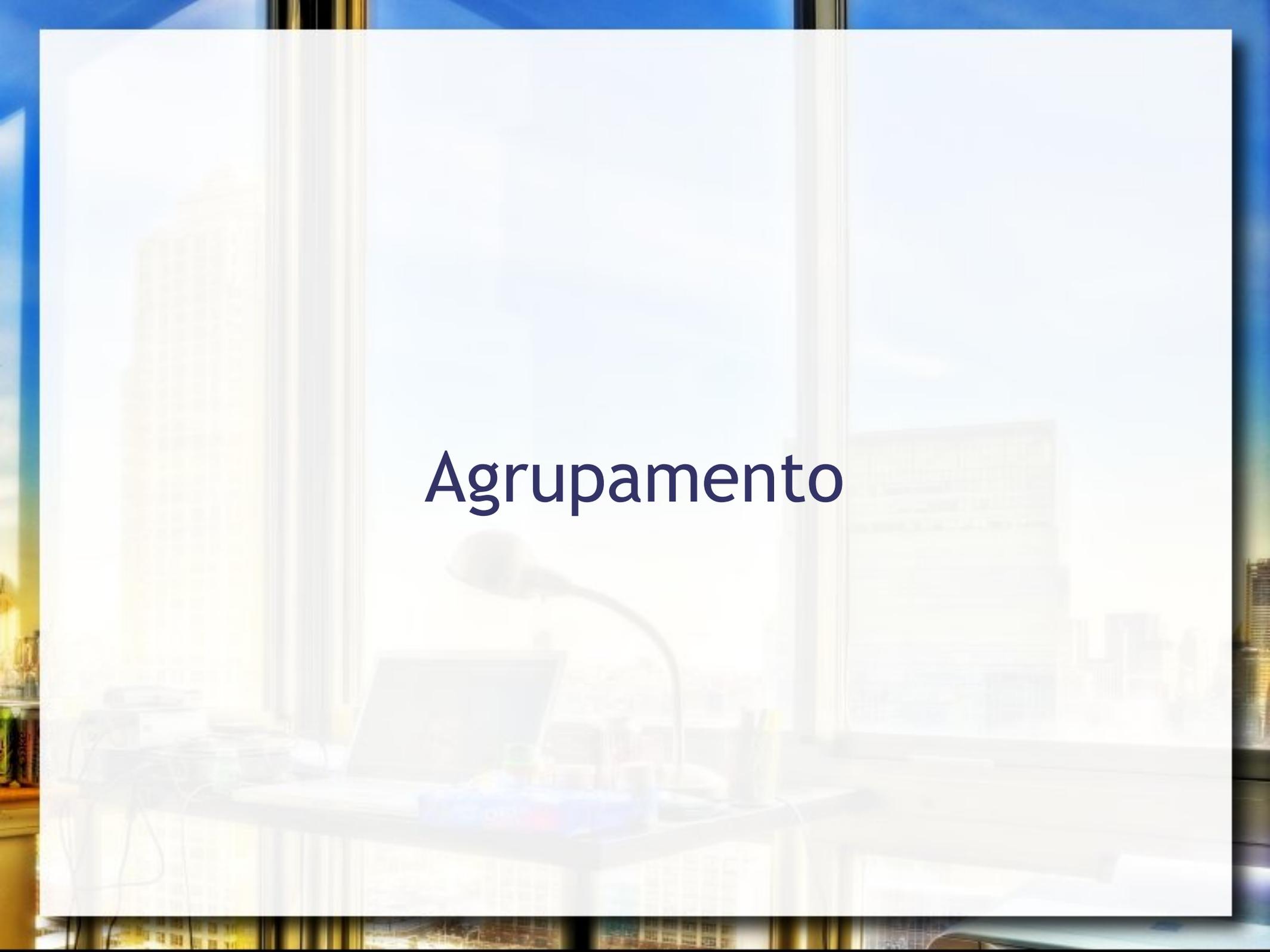
Utilizando o PreparedStatement

- INSERT INTO Marcadores
VALUES (? , ? , ? , ?)
- <comando>.setString(<numero>, <valor>)
- <comando>.setInt(<numero>, <valor>)

Utilizando o PreparedStatement

- UPDATE Marcadores
SET Categoria = ?
WHERE Categoria = ?
- <comando>.setString(<numero>, <valor>)
- <comando>.setInt(<numero>, <valor>)

Agrupamento

The image shows a blurred office desk in the foreground, featuring a laptop, a desk lamp, and various office supplies. The desk is positioned in front of a large window that offers a view of a city skyline with several tall buildings under a clear sky. The overall scene is brightly lit, suggesting a daytime setting. The text 'Agrupamento' is centered over the image in a dark blue, sans-serif font.

GROUP BY

- **SELECT * | <campo₁>[,..., <campo_n>]**
FROM <tabela₁>[,..., <tabela_n>]
WHERE <condição/junção>
GROUP BY <coluna_agrupar>
HAVING <condição_grupo>

Modelos de Taxi

```
SELECT T.Modelo
FROM Taxi T
GROUP BY T.Modelo;
```

Cliente (C)

<u>C liId</u>	N o m e	C P F
1532	A sdrúb al	448.754.253-65
1755	D oriana	567.387.387-44
1780	Q uincas	546.373.762-02



Táxi (TX)



<u>P la ca</u>	M ar ca	M o d e l o	A n o F a b
D A E 6534	Ford	Fiesta	1999
D K L 4598	W olkswagen	G ol	2001
D K L 7878	Ford	Fiesta	2001
J D M 8776	W olkswagen	S antana	2002
J J M 3692	C hevrolet	C orsa	1999



Corrida (R1)

<u>C liId</u>	<u>P la ca</u>	<u>D ata P e d i d o</u>
1755	D A E 6534	15/02/2003
1982	J D M 8776	18/02/2003

Funções de Agregação

- **COUNT(*)** ⇒ contagem
- **SUM(<coluna>)** ⇒ soma
- **AVG(<coluna>)** ⇒ média
- **MAX(<coluna>)** ⇒ maior valor
- **MIN(<coluna>)** ⇒ menor valor

Questão 5

Total de Taxis por Modelo

```
SELECT T.Modelo
FROM Taxi T
GROUP BY T.Modelo; (?)
```

Cliente (C)

<u>C liId</u>	N ome	C P F
1532	A sdrúb al	448.754.253-65
1755	D oriana	567.387.387-44
1780	Q uincas	546.373.762-02



Táxi (TX)



<u>P la ca</u>	M ar ca	M odelo	A no F ab
D A E 6534	Ford	Fiesta	1999
D K L 4598	W olkswagen	G ol	2001
D K L 7878	Ford	Fiesta	2001
J D M 8776	W olkswagen	S antana	2002
J J M 3692	Chevrolet	C orsa	1999



Corrida (R1)

<u>C liId</u>	<u>P la ca</u>	<u>D ata P edido</u>
1755	D A E 6534	15/02/2003
1982	J D M 8776	18/02/2003

Questão 6

Total de Taxis por Modelo

```

SELECT T.Modelo, COUNT (*)
FROM Taxi T
GROUP BY T.Modelo;
    
```

Cliente (C)

<u>C liId</u>	N o m e	C P F
1532	A sdrúb al	448.754.253-65
1755	D oriana	567.387.387-44
1780	Q uincas	546.373.762-02



Táxi (TX)



<u>P l a c a</u>	M a r c a	M o d e l o	A n o F a b
D A E 6534	Ford	Fiesta	1999
D K L 4598	W o l k s v a g e n	G o l	2001
D K L 7878	Ford	Fiesta	2001
J D M 8776	W o l k s v a g e n	S a n t a n a	2002
J J M 3692	C h e v r o l e t	C o r s a	1999



Corrida (R1)

<u>C liId</u>	<u>P l a c a</u>	<u>D a t a P e d i d o</u>
1755	D A E 6534	15/02/2003
1982	J D M 8776	18/02/2003

DISTINCT

Questão 6

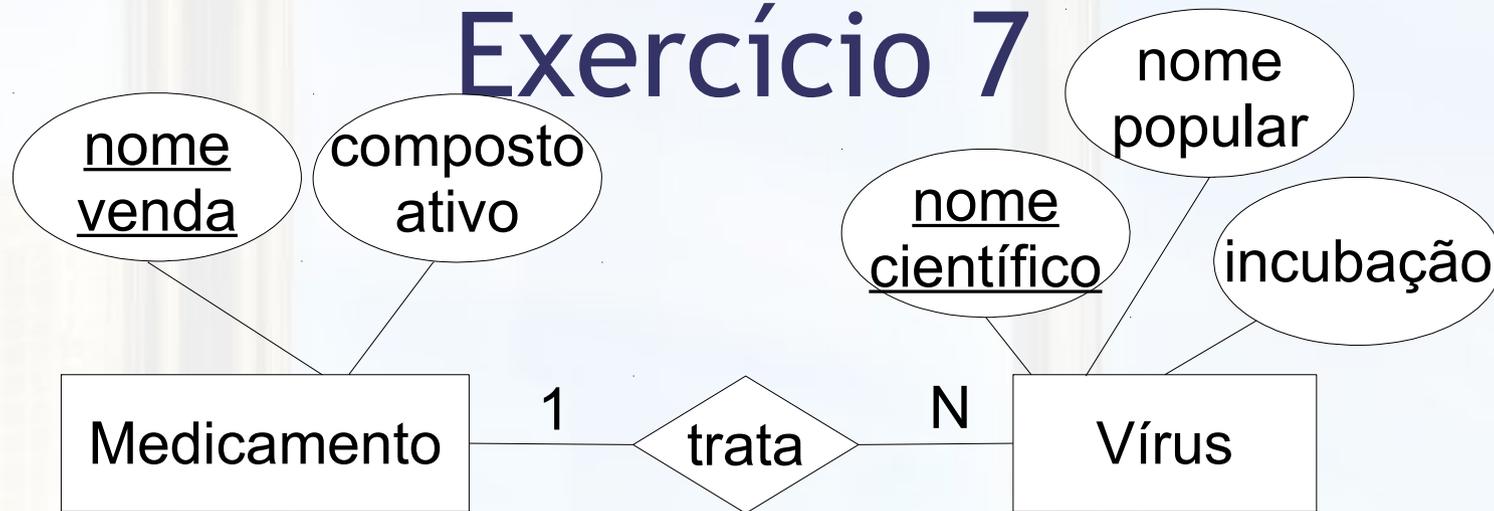
- Modelos de taxi tomados por cada cliente

```
SELECT DISTINCT Cl.Nome, T.Modelo
      FROM Cliente Cl, Corrida Co, Taxi T
      WHERE Cl.CliId = Co.CliId AND
            Co.Placa = T.Placa;
```

- Como fazer com GROUP BY?

```
SELECT DISTINCT Cl.Nome, T.Modelo
      FROM Cliente Cl, Corrida Co, Taxi T
      WHERE Cl.CliId = Co.CliId AND
            Co.Placa = T.Placa;
```

Exercício 7



medicamento (nomeVenda, compostoAtivo)

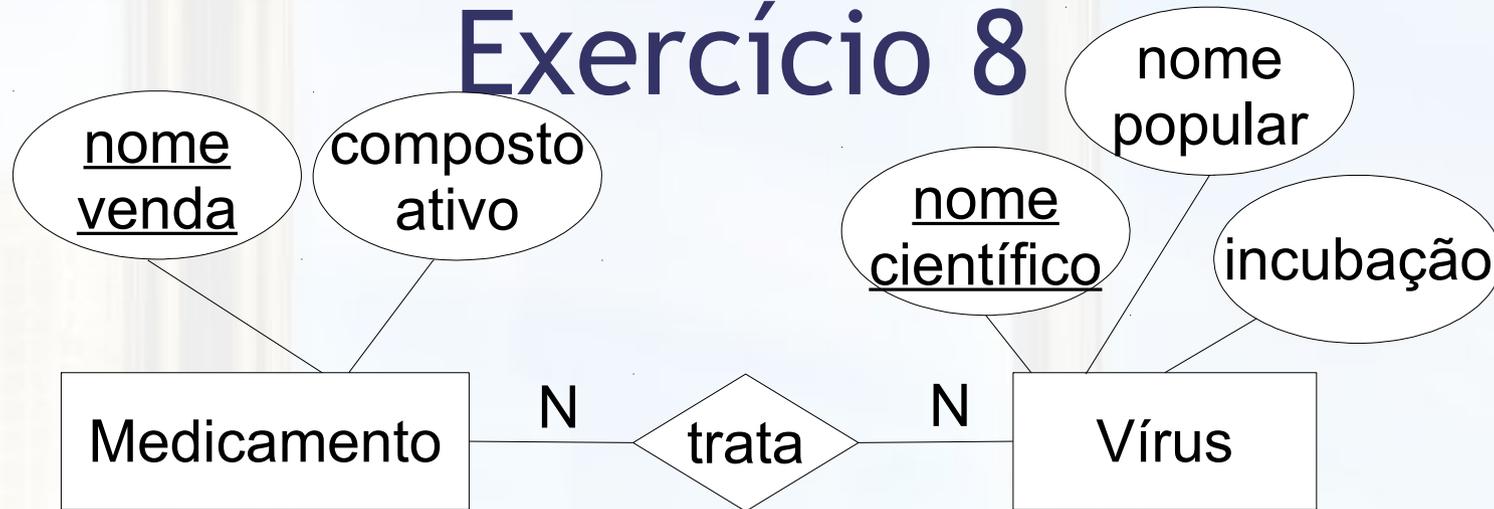
virus (nomeCientifico, nomePopular, incubacao, **nomeVendaMedicamento**)

- CHE: nomeVendaMedicamento para medicamento

a) Quantos vírus trata cada medicamento

b) Quantos vírus trata cada composto ativo

Exercício 8



medicamento (nomeVenda, compostoAtivo)

virus (nomeCientifico, nomePopular, incubacao)

trata (nomeVendaMedicamento, nomeCientificoVirus)

- CHE: nomeVendaMedicamento para medicamento

- CHE: nomeCientificoVirus para virus

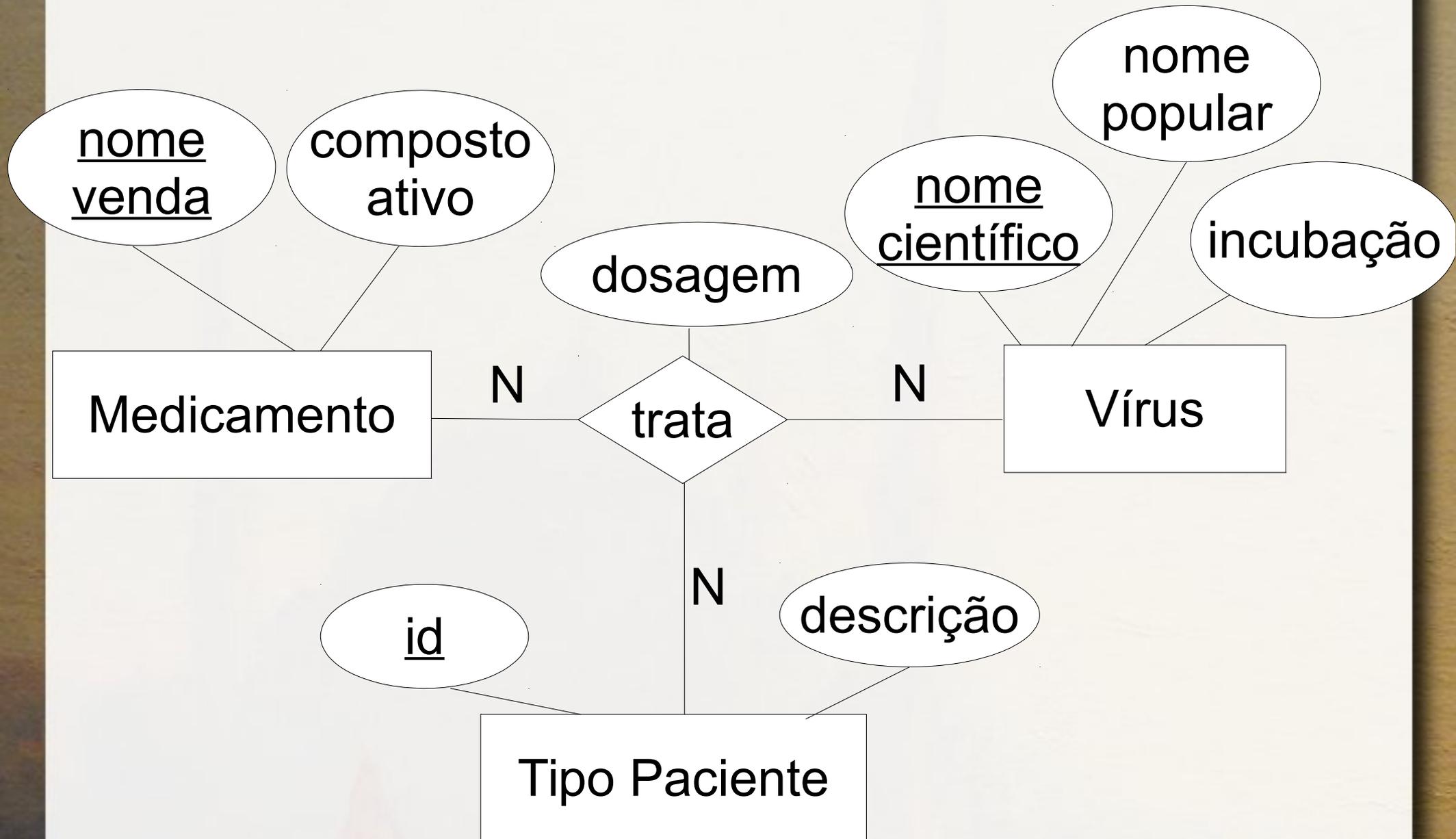
a) Quantos vírus trata cada medicamento

b) Quantos vírus trata cada composto ativo

Exercício para Casa 3

- Escreva uma sentença SQL, baseada no esquema abaixo, que retorne o número de pessoas da família em cada estado:
 - Pessoa(nome, nome_da_mãe, ano_nascimento, nome_cidade_natal)
 - nome_cidade_natal → CHE Cidade
 - Cidade(nome_cidade, sigla_estado)

Exercício para Casa 4



Exercício para Casa 4

Qual o vírus que exige maior dosagem de medicamento para tipoPaciente cuja descrição é 'idosos'

medicamento (nomeVenda, compostoAtivo)

virus (nomeCientifico, nomePopular, incubacao)

tipoPaciente (id, descricao)

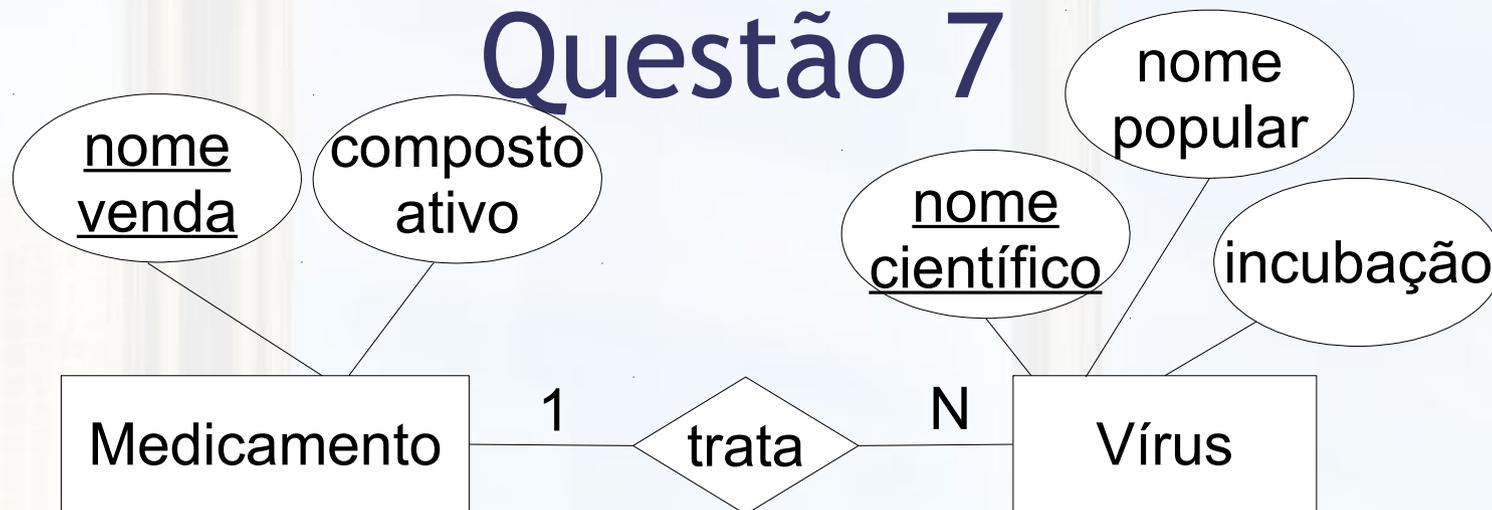
trata (nomeVendaMedicamento, nomeCientificoVirus, idTipoPaciente, dosagem)

- CHE: nomeVendaMedicamento para medicamento
- CHE: nomeCientificoVirus para virus
- CHE: idTipoPaciente para tipoPaciente

GROUP BY HAVING

- Testa a condição após o agrupamento
- WHERE - testa a condição antes do agrupamento
- ```
SELECT * | <campo1>[,..., <campon>]
FROM <tabela1>[,..., <tabelan>]
WHERE <condição/junção>
GROUP BY <coluna_agrupar>
HAVING <condição_grupo>
```

# Questão 7

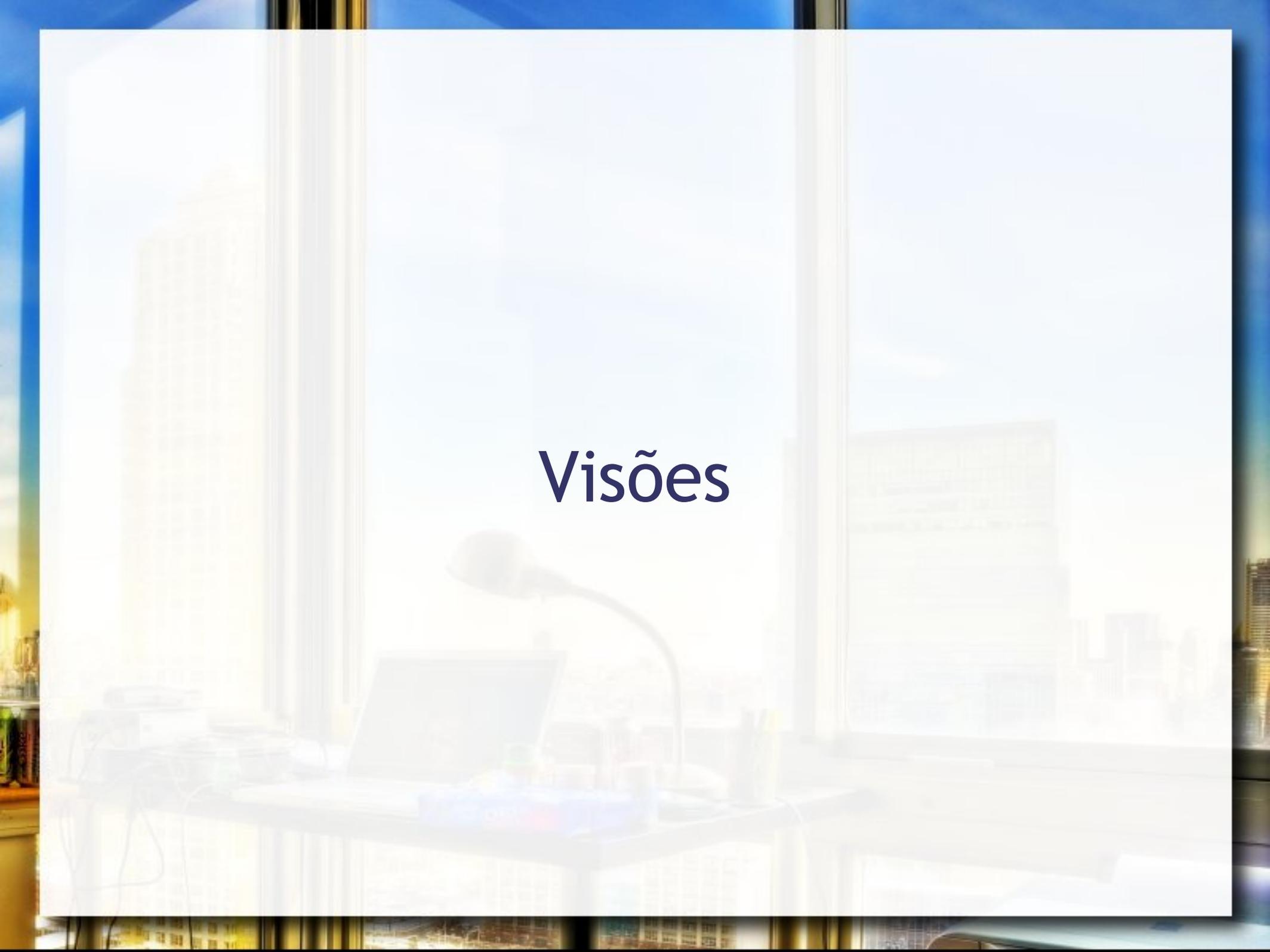


medicamento (nomeVenda, compostoAtivo)

virus (nomeCientifico, nomePopular, incubacao, **nomeVendaMedicamento**)

- CHE: nomeVendaMedicamento para medicamento

- Quais os compostos ativos que tratam vírus com período de incubação maior que 5 dias
- Quais os compostos ativos que tratam mais que 5 vírus

A blurred office desk with a laptop and a desk lamp, viewed through a window overlooking a city skyline. The scene is dimly lit, suggesting an evening or early morning setting. The desk is cluttered with various items, including a laptop, a desk lamp, and some papers. The window provides a view of a city with several tall buildings under a cloudy sky.

# Visões

# VIEW

- **CREATE VIEW** <nome> **AS**  
**SELECT ...**

# VIEW

## Questão 8

- Criar uma tabela de Modelos de Taxi com o nome do Modelo e o número de Taxis por modelo
- **CREATE VIEW** <nome> **AS**  
**SELECT ...**

# VIEW

## Questão 8

- Criar uma tabela de Modelos de Taxi com o nome do Modelo e o número de Taxis por modelo
- ```
CREATE VIEW Contagem_Modelo AS  
SELECT Modelo, COUNT(*)  
Numero_Taxis  
FROM taxi  
GROUP BY Modelo;
```



Consultas Aninhadas

SELECT Seleção

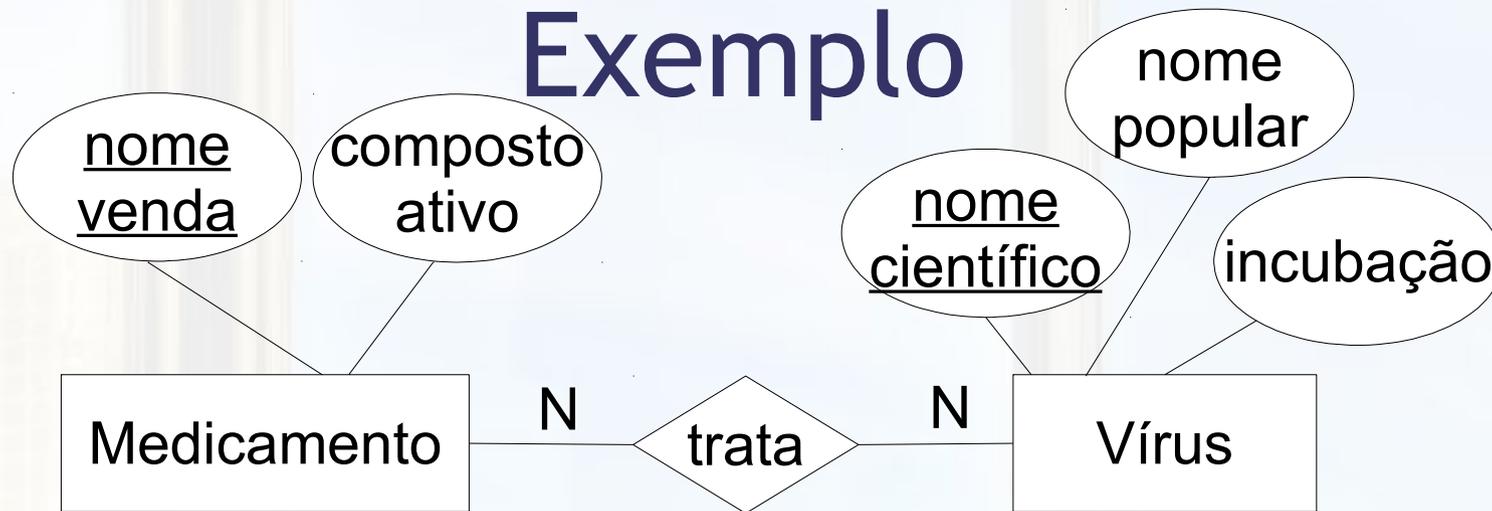
SELECT * FROM Taxi WHERE AnoFab > 2000

<u>Placa</u>	Marca	Modelo	AnoFab
D K L 4 5 9 8	W o l k s v a g e n	G o l	2 0 0 1
D K L 7 8 7 8	F o r d	F i e s t a	2 0 0 1
J D M 8 7 7 6	W o l k s v a g e n	S a n t a n a	2 0 0 2

SELECT IN e NOT IN

- SELECT ...
WHERE <campo> IN
(SELECT <campo> ...)
- SELECT ...
WHERE <campo> NOT IN
(SELECT <campo> ...)

Exemplo



medicamento (nomeVenda, compostoAtivo)

virus (nomeCientifico, nomePopular, incubacao)

trata (nomeVendaMedicamento, nomeCientificoVirus)

- CHE: nomeVendaMedicamento para medicamento
- CHE: nomeCientificoVirus para virus
- Nome popular dos vírus trata o medicamento com composto ativo Virulex

SELECT EXISTS e NOT EXISTS

- SELECT ...
WHERE EXISTS
(SELECT <campo> ...)
- SELECT ...
WHERE NOT EXISTS
(SELECT <campo> ...)

SELECT Comparação

- SELECT ...
WHERE <campo> <comparação>
(SELECT <campo> ...)

Exercício para Casa 5

- Para as tabelas que você montou no exercício 1, escreva um comando SQL que retorne todos os primos por parte de mãe, que você for capaz de inferir a partir da tabela. Considere que você tem como ponto de partida o nome da sua avó.
- Utilize duas estratégias:
 - VIEW
 - SELECT aninhado

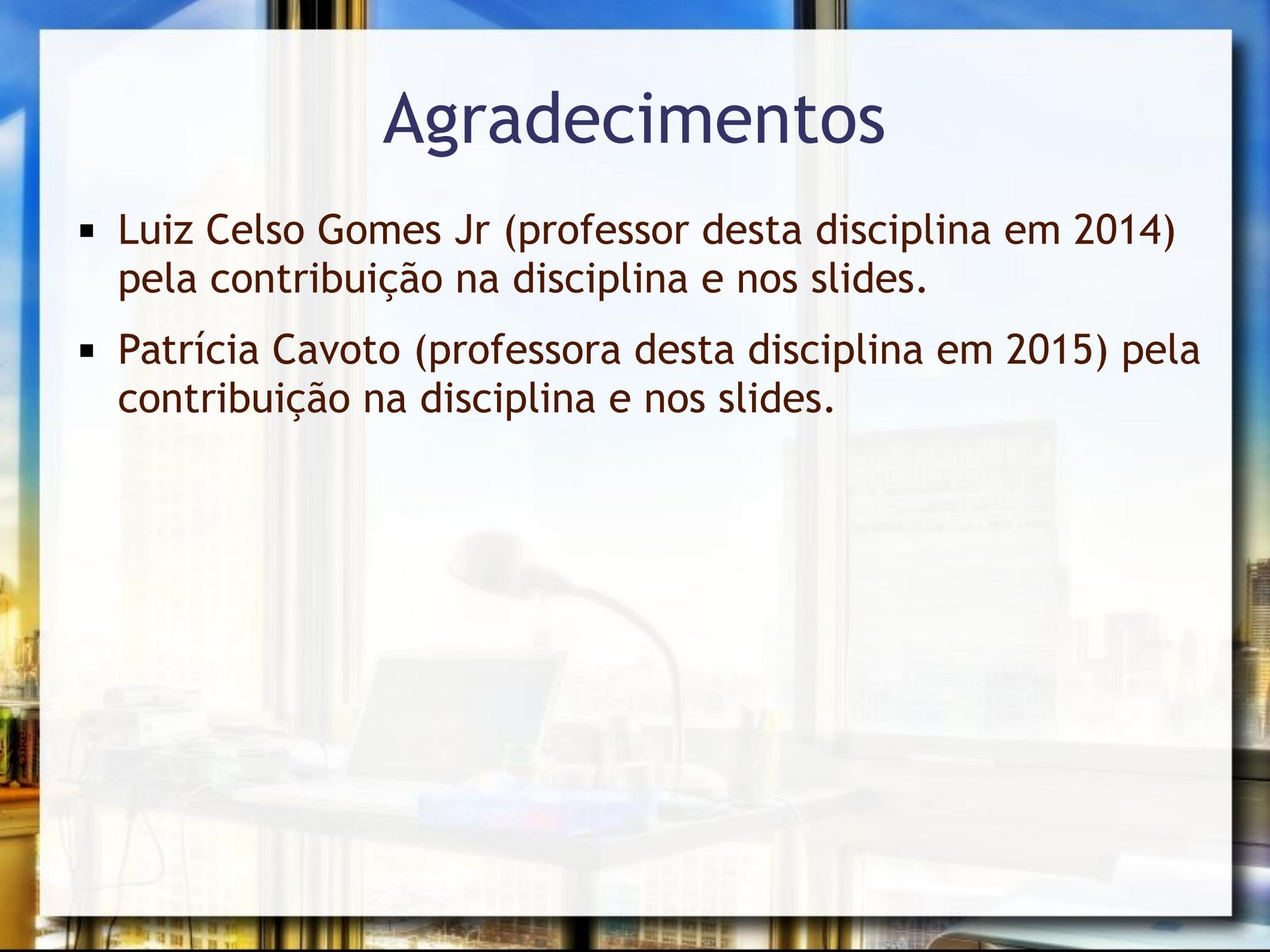
SELECT aninhado também pode ser usado em operações de **UPDATE** e **DELETE**

União, Interseção e Diferença

- **SELECT ...**
<operador>
SELECT ...
- **<operador>**
 - **UNION**
 - **INTERSECT**
 - **EXCEPT**

Agradecimentos

- Luiz Celso Gomes Jr (professor desta disciplina em 2014) pela contribuição na disciplina e nos slides.
- Patrícia Cavoto (professora desta disciplina em 2015) pela contribuição na disciplina e nos slides.



André Santanchè

<http://www.ic.unicamp.br/~santanche>

Licença

- Estes slides são concedidos sob uma Licença Creative Commons. Sob as seguintes condições: Atribuição, Uso Não-Comercial e Compartilhamento pela mesma Licença, com restrições adicionais:
 - Se você é estudante, você não está autorizado a utilizar estes slides (total ou parcialmente) em uma apresentação na qual você esteja sendo avaliado, a não ser que o professor que está lhe avaliando:
 - lhe peça explicitamente para utilizar estes slides;
 - ou seja informado explicitamente da origem destes slides e concorde com o seu uso.
- Mais detalhes sobre a referida licença Creative Commons veja no link:
<http://creativecommons.org/licenses/by-nc-sa/2.5/br/>