

Aplicações - SQL

Banco de Dados: Teoria e Prática

André Santanchè
Instituto de Computação – UNICAMP
Setembro de 2019



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 de dados
- 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

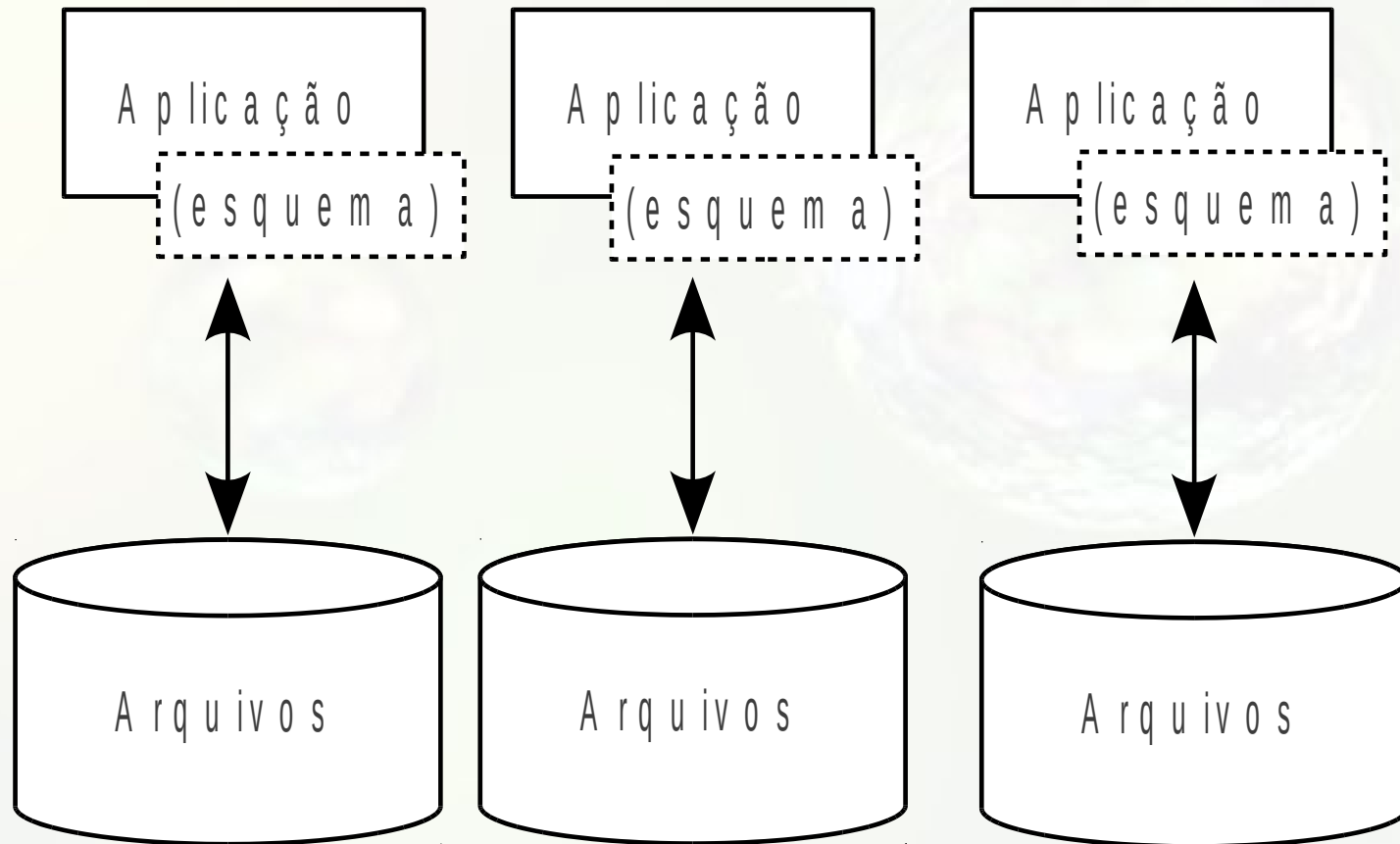
Padronização

- ANSI + ISO
- SQL-86 ou SQL1
- SQL-92 ou SQL2
- SQL:1999 ou SQL3
- SQL:2003
- SQL:2006
- SQL:2008
- SQL:2011
- SQL:2016

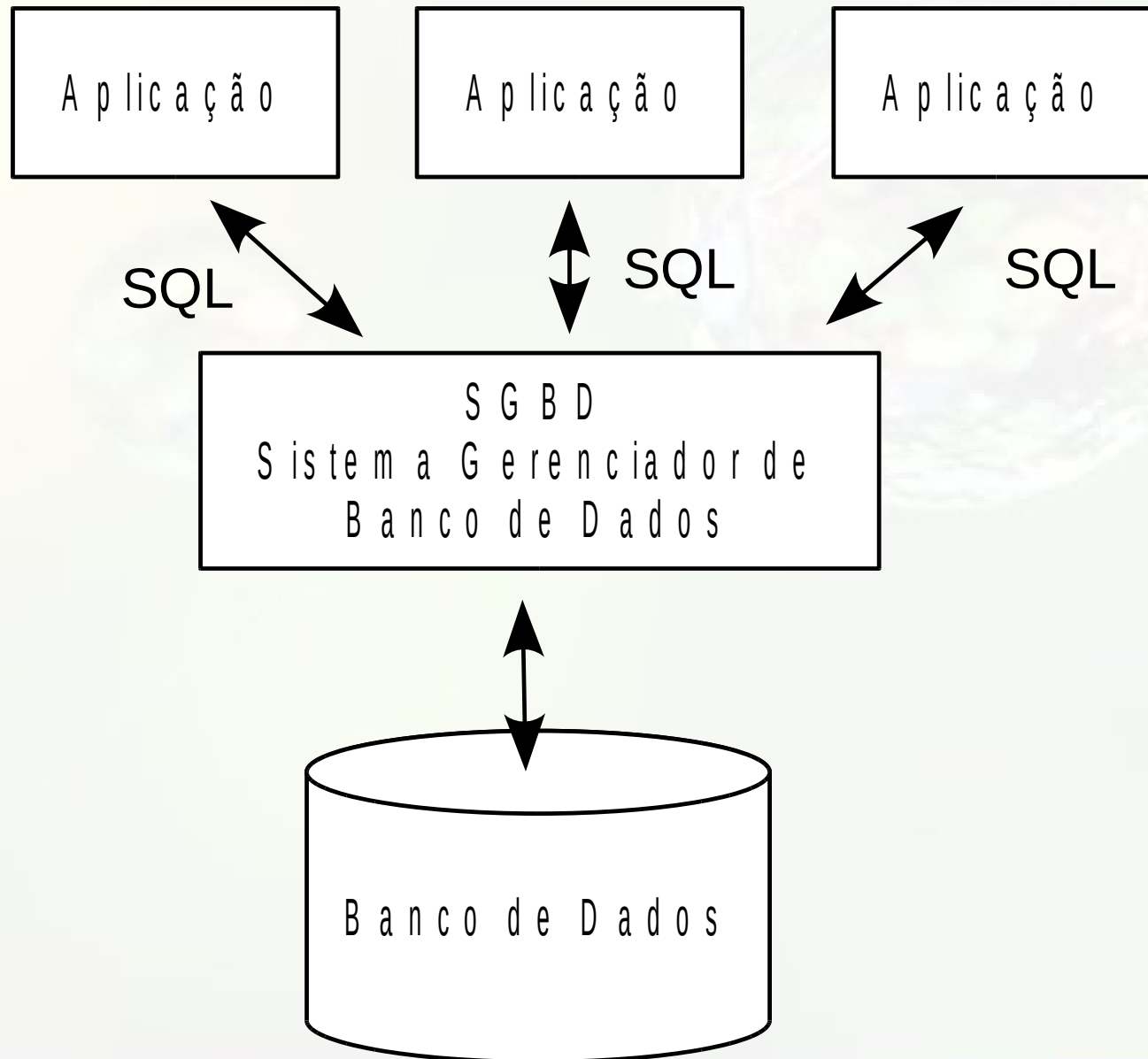
SQL Padronização

- ANSI + ISO
- SQL-86 ou SQL1
- SQL-92 ou SQL2
- SQL:1999 ou SQL3
- SQL:2003
- **SQL:2006 → adotaremos essa versão**
- SQL:2008
- SQL:2011
- SQL:2016

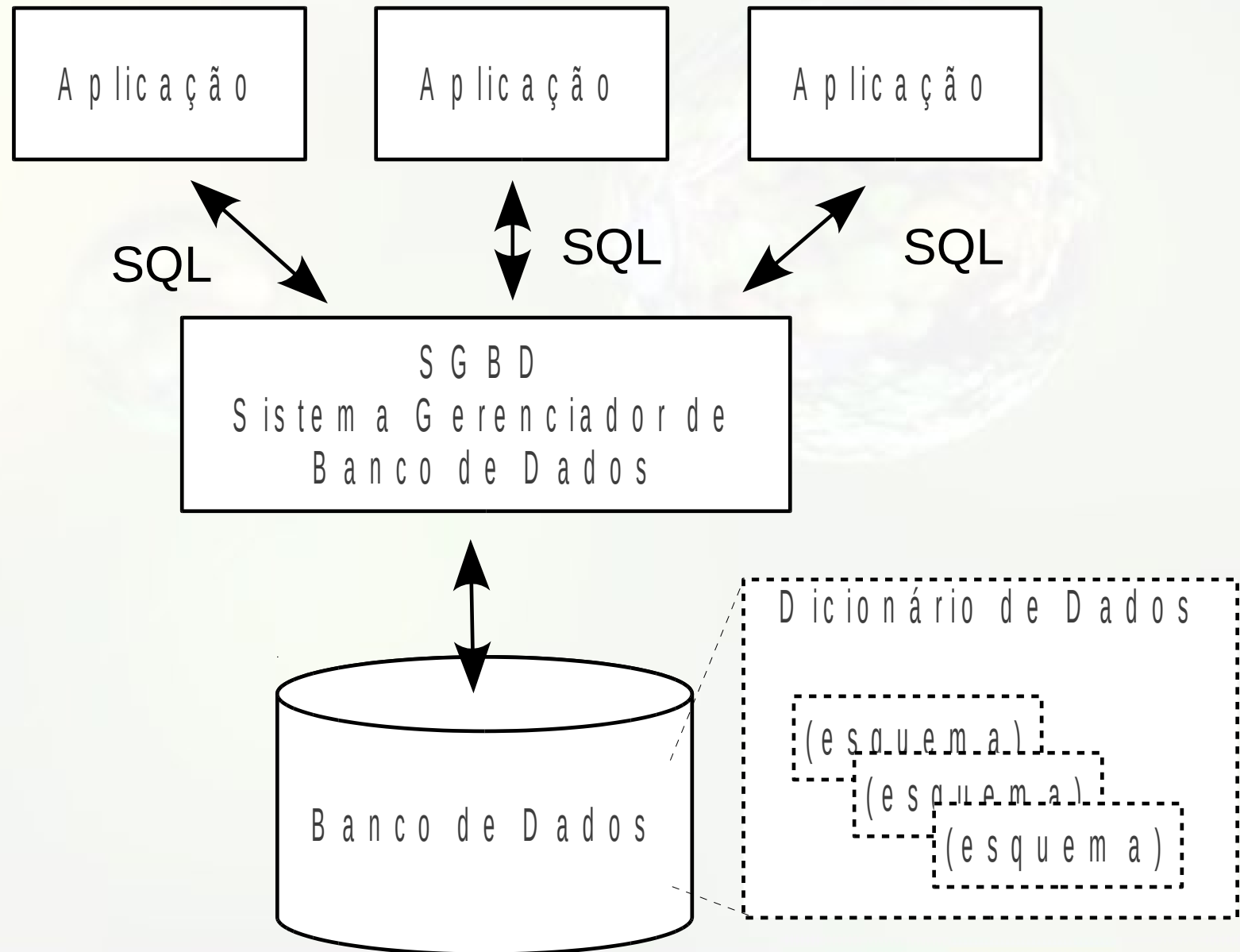
Aplicações e Armazenamento Arquivos



Aplicações e Armazenamento SGBD



Dicionário de Dados



Caso Prático - Taxis

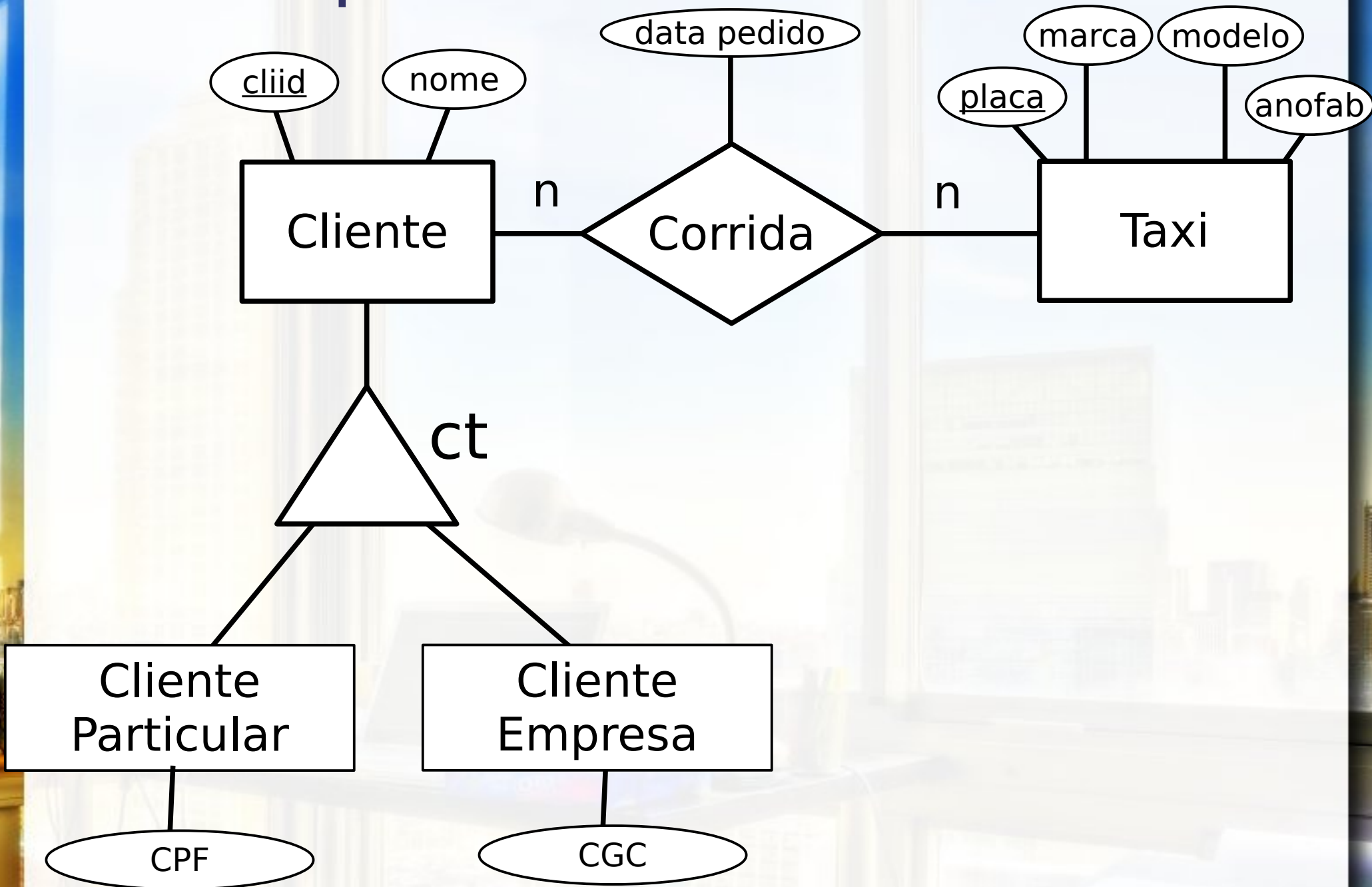
The background of the slide is a blurred office scene. In the foreground, a desk is visible with a laptop, a desk lamp, and some papers. The desk is positioned in front of a large window that offers a view of a city skyline with several tall buildings. The overall lighting is bright, suggesting a daytime setting. The text 'Caso Prático - Taxis' is centered over the image in a dark blue, sans-serif font.

Esquema Conceitual

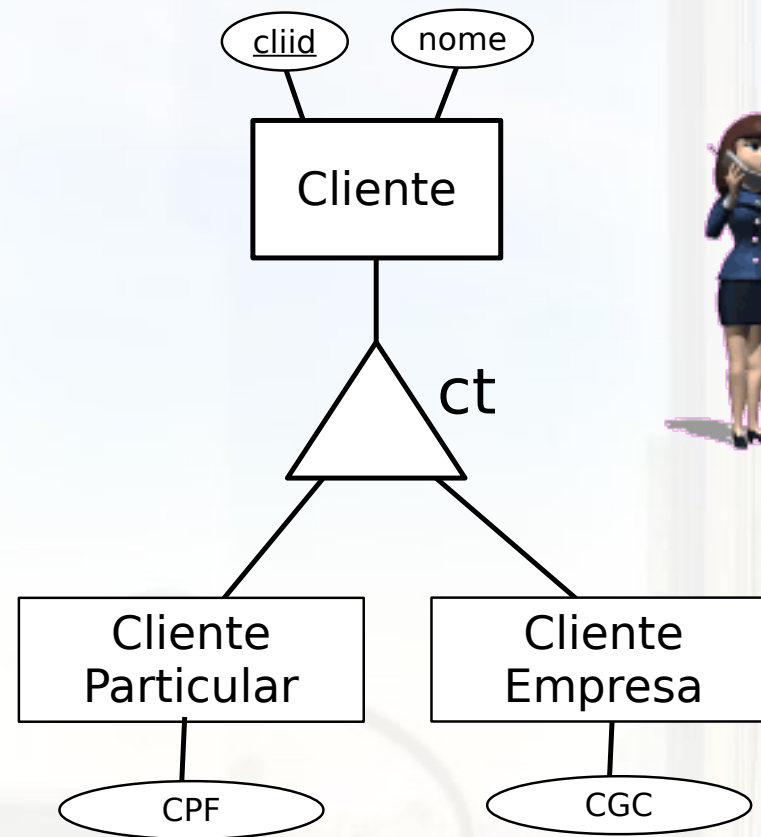
Exemplo Táxis

- 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 Taxis



Esquema Conceitual Exemplo Táxis



Cliente Particular - indivíduo que irá pagar a conta

Cliente Empresa - funcionário de uma empresa conveniada, para a qual a conta será enviada

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

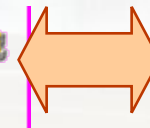
Táxi (TX)

<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>C l i d</u>	<u>Placa</u>	<u>Data Pedido</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][DEFAULT <valor>] [restrição],
[...,
<campo_n> <tipo> [NULL|NOT NULL][DEFAULT <valor>] [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)  
);
```

Tipos de Dados

- VARCHAR(<máximo de caracteres>) - String
- Tipos numéricos exatos
 - SMALLINT - inteiro pequeno
 - INTEGER - inteiro
 - DECIMAL(<máximo de dígitos>, <decimais>) - número com casas decimais fixas
- Tipos numéricos aproximados
 - REAL
 - DOUBLE PRECISION

(precisão dos tipos numéricos são definidas pela implementação)

- DATE
- TIME

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)

```
CREATE TABLE <tabela>  
  (<campo1> <tipo> [NULL|NOT NULL] [restrição],  
  [...,  
  <campon> <tipo> [NULL|NOT NULL] ]  
  PRIMARY KEY (<chave_primaria>))
```

```
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, CASCADE, SET NULL, SET DEFAULT

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.

```
INSERT INTO <tabela>  
  [(<campo1>[,..., <campon>])]  
  VALUES ( <valor1>[,..., <valorn>])
```

SELECT

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

- `executeQuery(...)`

SELECT Projeção

SELECT Marca, Modelo FROM Taxi

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

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	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	Corso	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 * | <campo1>[, ..., <campon>]  
FROM <tabela1>[, ..., <tabelan>]  
WHERE <condição/junção>
```

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

- 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 _

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 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

(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

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 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

(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

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 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

(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) 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
1532	A s d r ú b a l	1755	D A E 6 5 3 4	15/02/2003
1532	A s d r ú b a l	1982	J D M 8 7 7 6	18/02/2003
1755	D o r i a n a	1755	D A E 6 5 3 4	15/02/2003
1755	D o r i a n a	1982	J D M 8 7 7 6	18/02/2003
1780	Q u i n c a s	1755	D A E 6 5 3 4	15/02/2003
1780	Q u i n c a s	1982	J D M 8 7 7 6	18/02/2003

Join (1)

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

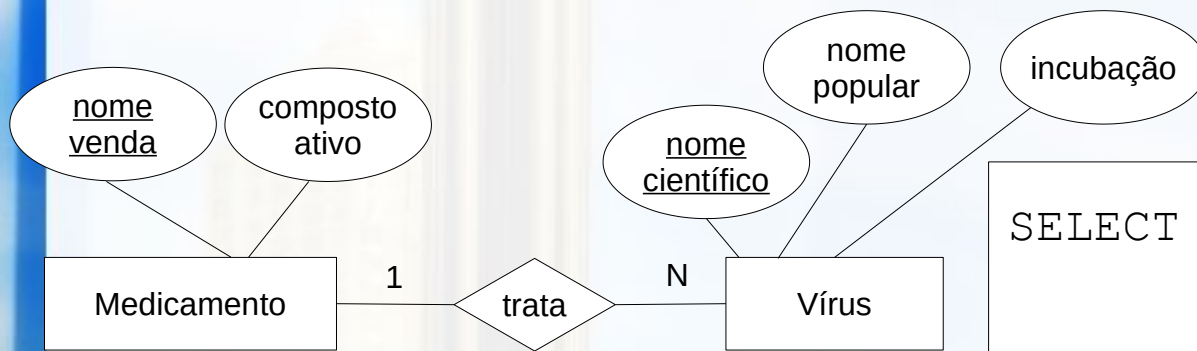
(C liId)	N o m e	(C lId)	P l a c a	D a t a P e d i d o
1532	A s d r ú b a l	1755	D A E 6 5 3 4	15/02/2003
1532	A s d r ú b a l	1982	J D M 8 7 7 6	18/02/2003
1755	D o r i a n a	1755	D A E 6 5 3 4	15/02/2003
1755	D o r i a n a	1982	J D M 8 7 7 6	18/02/2003
1780	Q u i n c a s	1755	D A E 6 5 3 4	15/02/2003
1780	Q u i n c a s	1982	J D M 8 7 7 6	18/02/2003

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 lId)	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 4



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

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

Questão

- 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

- Modelos de taxi existentes

- Sem DISTINCT

```
SELECT Modelo  
FROM Taxi;
```

- Com DISTINCT

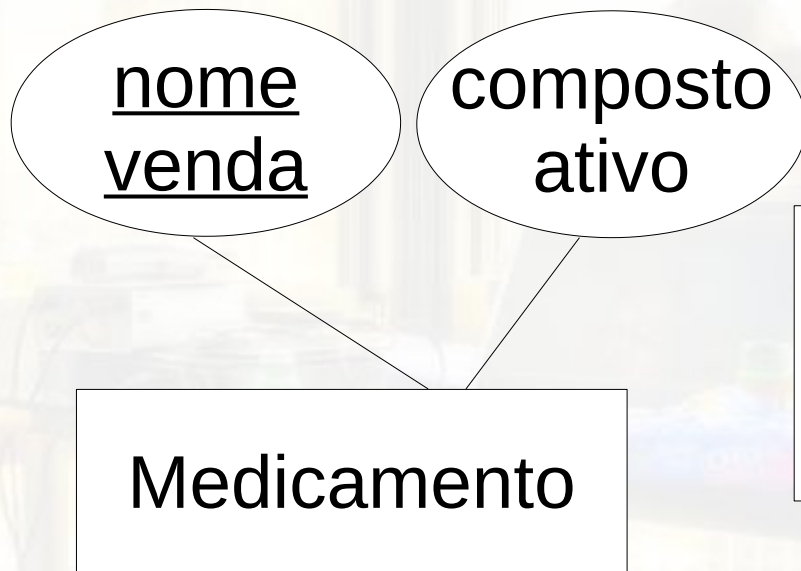
```
SELECT DISTINCT Modelo  
FROM Taxi;
```

Exercício 5

■ Liste todos os compostos ativos disponíveis

■ Esquema:

□ medicamento(nomeVenda, compostoAtivo)



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

Questão

■ Qual o modelo de Taxi para cada Corrida?

Cliente (C)

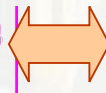
<u>C liId</u>	N o m e	C P F
1532	A sdrúb al	448.754.253-65
1755	D oriana	567.387.387-44
1780	Q uinc as	546.373.762-02



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	Corso	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 Corrida.DataPedido, Corrida.Placa, Taxi.Modelo  
FROM Corrida, Taxi  
WHERE Corrida.Placa = Taxi.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

Desafio

- Quais os modelos de Taxi tomados por cada Cliente? (clientes identificados pelo nome)

Cliente (C)

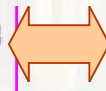
<u>C liId</u>	N o m e	C P F
1532	A sdrúb al	448.754.253-65
1755	D oriana	567.387.387-44
1780	Q uinc as	546.373.762-02



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 ataP edido</u>
1755	D A E 6534	15/02/2003
1982	J D M 8776	18/02/2003

Modelos de Táxi por Cliente (estágio 1)

```

SELECT Cliente.Nome, Corrida.DataPedido, Corrida.Placa, Taxi.Modelo
FROM Cliente, Corrida, Taxi
WHERE Cliente.CliId = Corrida.CliId AND Corrida.Placa = Taxi.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

Modelos de Táxi por Cliente (estágio 2)

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

AS

alias (apelido)

- SELECT <campo₁> [AS] <alias₁>
[,..., <campo_n> [AS] <alias_n>]

...

- SELECT ...
FROM <tabela₁> [AS] <alias₁>
[,..., <tabela_n> [AS] <alias_n>]

...

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;
```

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)

Mesma Tabela com dois ALIAS

- A mesma tabela pode aparecer em dois pontos a consulta com nomes diferentes como se fossem duas tabelas, cada uma com um ALIAS

Mesma Tabela com dois ALIAS

Questão - parte 1

- Para as tabelas que você montou no exercício 1, escreva um comando SQL que retorne:
 - nomes das pessoas que nasceram na mesma cidade de Asdrúbal
- Esquema
 - Pessoa(nome, nome_da_mãe, ano_nascimento, nome_cidade_natal)
 - nome_cidade_natal → CHE Cidade
 - Cidade(nome_cidade, sigla_estado)

Questão - parte 1

■ Nomes das pessoas que nasceram na mesma cidade de Asdrúbal

■ Esquema

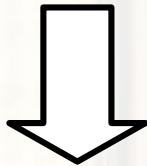
□ Pessoa(nome, nome_da_mãe, ano_nascimento, nome_cidade_natal)

◦ nome_cidade_natal → CHE Cidade

□ Cidade(nome_cidade, sigla_estado)

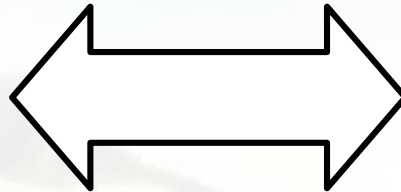
```
■ SELECT P2.nome
FROM Pessoa P1, Pessoa P2
WHERE P1.nome = 'Asdrúbal' and
      P2.nome_cidade_natal=P1.nome_cidade_natal and
      P2.nome <> 'Asdrúbal'
```


Pessoa: P1



	Asdrúbal

nome_cidade_natal = nome_cidade_natal



Pessoa: P2



	<> Asdrúbal

```

SELECT P2.nome
FROM Pessoa P1, Pessoa P2
WHERE P1.nome = 'Asdrúbal' and
      P2.nome_cidade_natal=P1.nome_cidade_natal and
      P2.nome <> 'Asdrúbal'

```

Mesma Tabela com dois ALIAS

Questão - parte 2

- Para as tabelas que você montou no exercício 1, escreva um comando SQL que retorne:
 - nomes das pessoas que nasceram no mesmo estado de Asdrúbal
- Esquema
 - Pessoa(nome, nome_da_mãe, ano_nascimento, nome_cidade_natal)
 - nome_cidade_natal → CHE Cidade
 - Cidade(nome_cidade, sigla_estado)

Questão - parte 2

■ nomes das pessoas que nasceram no mesmo estado de Asdrúbal.

■ Esquema

□ Pessoa(nome, nome_da_mãe, ano_nascimento, nome_cidade_natal)

◦ nome_cidade_natal → CHE Cidade

□ Cidade(nome_cidade, sigla_estado)

■ SELECT P2.nome

FROM Pessoa P1, Cidade C1, Pessoa P2, Cidade C2

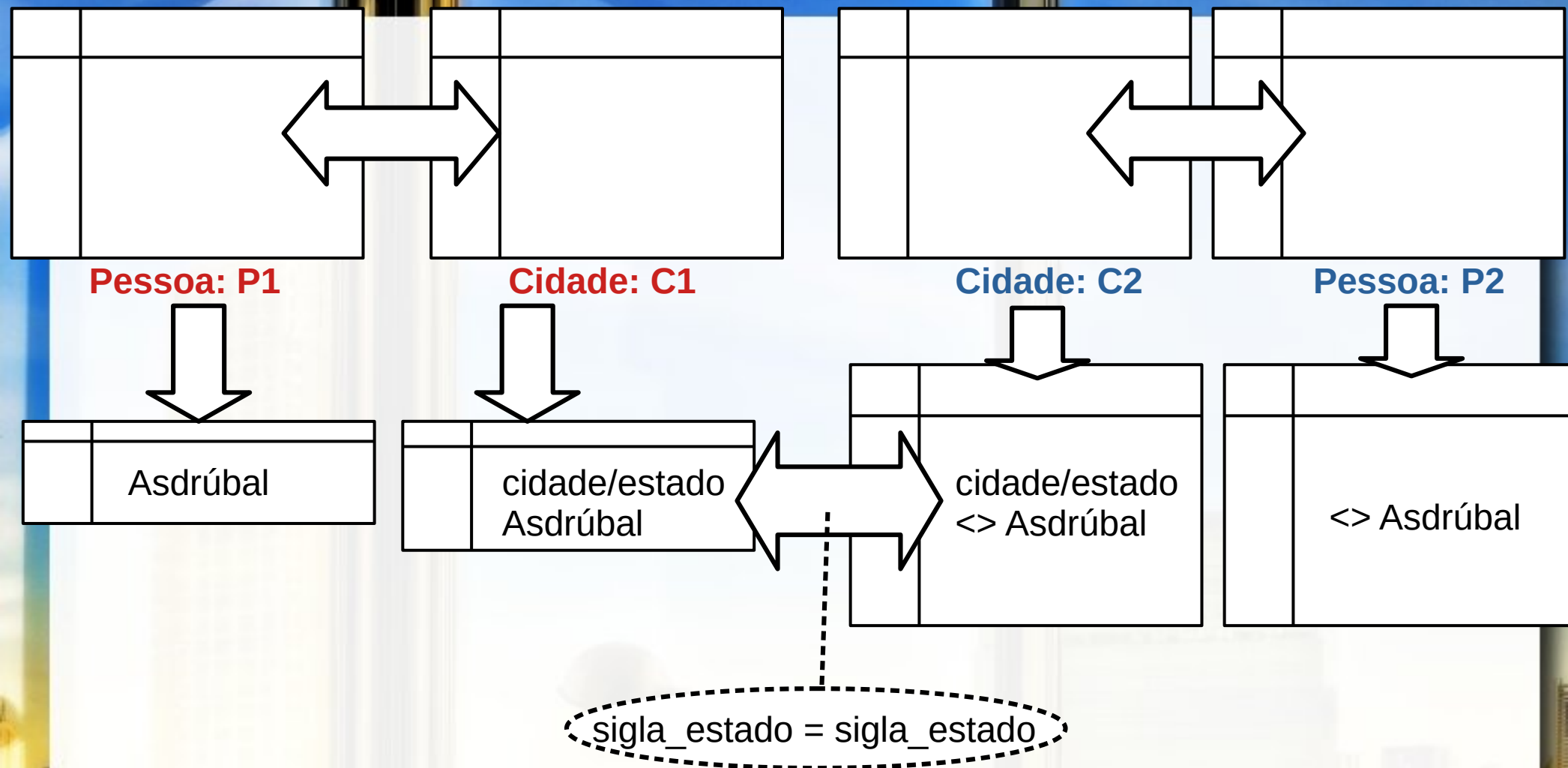
WHERE P1.nome = 'Asdrúbal' and

P1.nome_cidade_natal = C1.nome_cidade and

P2.nome <> 'Asdrúbal' and

P2.nome_cidade_natal = C2.nome_cidade and

C1.sigla_estado = C2.sigla_estado



```

SELECT P2.nome
FROM Pessoa P1, Cidade C1, Pessoa P2, Cidade C2
WHERE P1.nome = 'Asdrúbal' and
      P1.nome_cidade_natal = C1.nome_cidade and
      P2.nome <> 'Asdrúbal' and
      P2.nome_cidade_natal = C2.nome_cidade and
      C1.sigla_estado = C2.sigla_estado

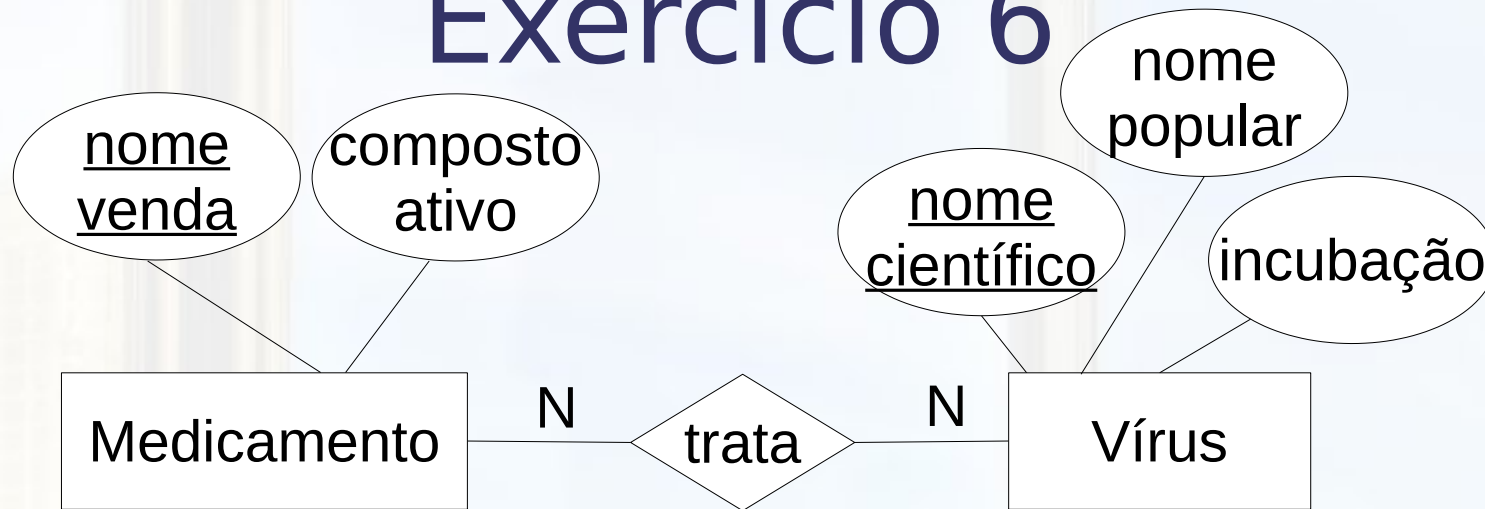
```

Mesma Tabela com dois ALIAS

Exercício para Casa

- Para as tabelas que você montou no exercício 1, escreva um comando SQL que retorne:
 - retorne todos os primos por parte de mãe, que você for capaz de inferir a partir da tabela
- Esquema
 - Pessoa(nome, nome_da_mãe, ano_nascimento, nome_cidade_natal)
 - nome_cidade_natal → CHE Cidade
 - Cidade(nome_cidade, sigla_estado)

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

SELECT ORDER BY

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

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

Questão ORDER BY

■ Cliente e os modelos de taxi tomados?

Questão

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 T.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

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 T.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

ORDER BY

■ Cliente e os modelos de taxi tomados

■ ?

```
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;
```

■ ?

```
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;
```

Questão

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;
```

A blurred office desk with a laptop, a desk lamp, and a window overlooking a city skyline. The text "Join (2) explícito" is centered on the image.

Join (2) explícito

Join (2)

- **SELECT ...**
FROM <tabela> JOIN <tabela>
ON <condição> ...
- **Tipo clássico de join explícito**

Join (2)

- Taxis e respectivas corridas (para taxis que fizeram corrida)

- Implícito:

```
SELECT Tx.placa, Co.cliid  
FROM Taxi Tx, Corrida Co  
WHERE Tx.placa = Co.placa;
```

- Explícito:

```
SELECT Tx.placa, Co.cliid  
FROM Taxi Tx JOIN Corrida Co  
ON Tx.placa = Co.placa;
```

Natural Join

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

Join (2)

- Taxis e respectivas corridas (para taxis que fizeram corrida)
- ```
SELECT Tx.placa, Co.cliid
FROM Taxi Tx
 NATURAL JOIN Corrida Co;
```

# 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



# Join (2)

- Taxis e respectivas corridas (para todos os taxis)
- ```
SELECT Tx.placa, Co.cliid  
FROM Taxi Tx LEFT JOIN Corrida Co  
ON Tx.placa = Co.placa;
```

Join (2)

- Corridas e respectivos clientes (para todos os clientes)
- ```
SELECT Co.placa, Cl.nome
FROM Corrida Co RIGHT JOIN Cliente Cl
ON Co.cliid = Cl.cliid;
```

# DELETE

- DELETE FROM <tabela<sub>1</sub>>  
WHERE <condição>
- executeUpdate(...)

# Excluir as Corridas feitas pelo Taxi DAE6534

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

## Cliente (C)

| <u>CliId</u> | Nome     | CPF            |
|--------------|----------|----------------|
| 1532         | Asdrúbal | 448.754.253-65 |
| 1755         | Doriana  | 567.387.387-44 |
| 1780         | Quincas  | 546.373.762-02 |



## Táxi (TX)



| <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   |



## Corrida (R1)

| <u>ClId</u> | <u>Placa</u> | <u>DataPedido</u> |
|-------------|--------------|-------------------|
| 1755        | DAE6534      | 15/02/2003        |
| 1982        | JDM8776      | 18/02/2003        |

# Violação de Integridade

- Excluir Táxis que possuam corridas

```
DELETE FROM Taxi WHERE Taxi.Placa =
'DKL4598';
```

# UPDATE

- UPDATE <tabela>  
SET <campo<sub>1</sub>>=<valor<sub>1</sub>>  
[,..., <campo<sub>n</sub>>=<valor<sub>n</sub>>]  
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>ClId</u> | Nome     | CPF            |
|-------------|----------|----------------|
| 1532        | Asdrúbal | 448.754.253-65 |
| 1755        | Doriana  | 567.387.387-44 |
| 1780        | Quincas  | 546.373.762-02 |



## Táxi (TX)



| <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   |



## Corrida (R1)

| <u>ClId</u> | <u>Placa</u> | <u>DataPedido</u> |
|-------------|--------------|-------------------|
| 1755        | DAE6534      | 15/02/2003        |
| 1982        | JDM8776      | 18/02/2003        |



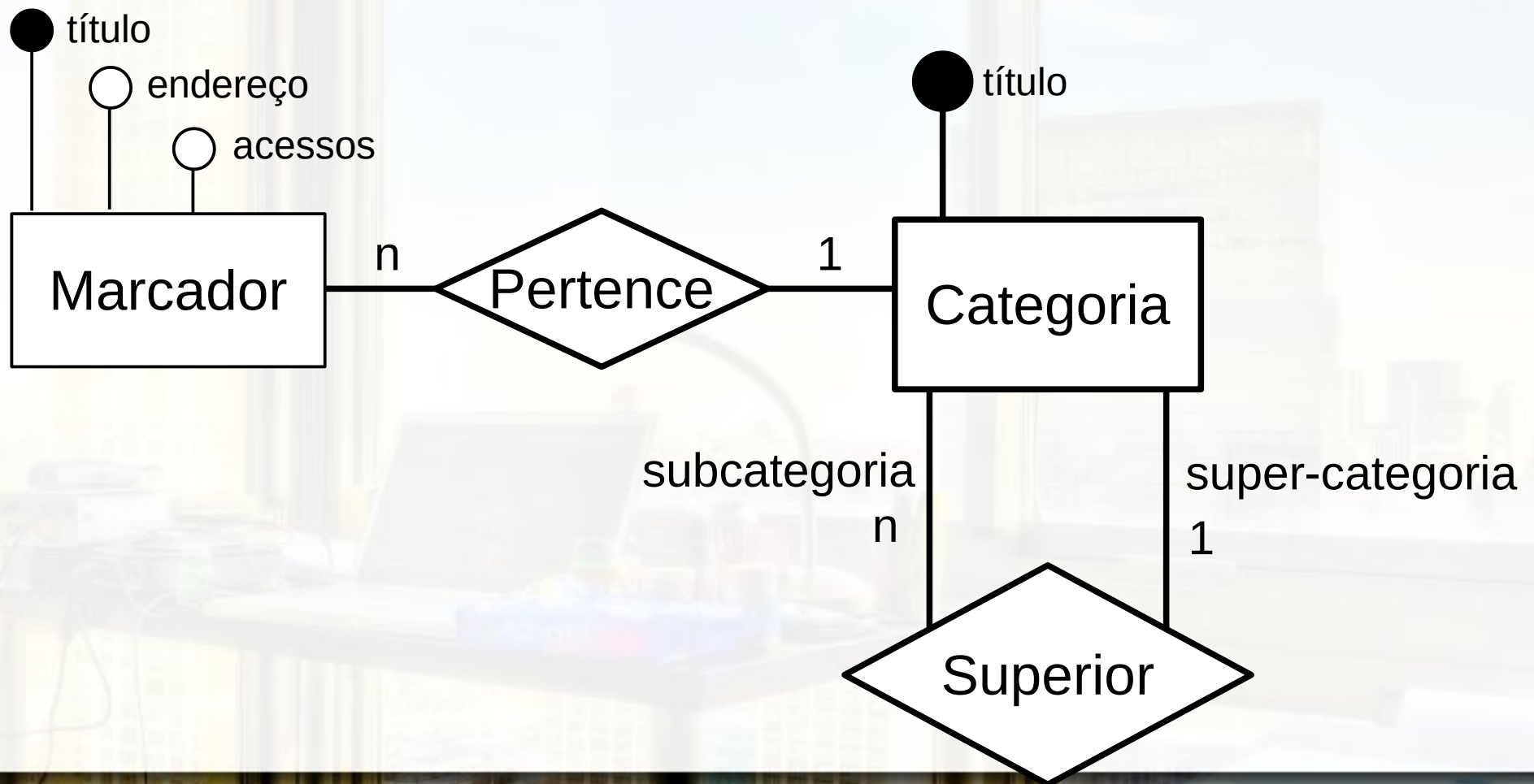
# UPDATE em mais de uma Tabela



# Questão

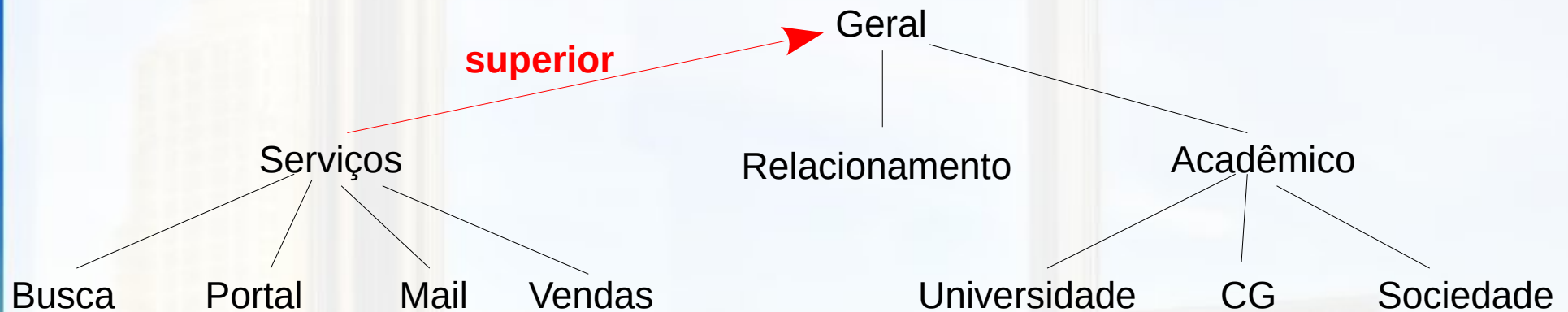
## Marcadores e Categorias

- Considere o modelo de Marcadores e Categorias



# Questão

## Marcadores e Categorias



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

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

# Exercício

## Marcadores e Categorias

- Como mudar o título de uma Categoria já que há Marcadores que dependem dela?

Marcador(Título, Acessos, Endereço, Categoria)

- Categoria: chave estrangeira  
para Taxonomia

Taxonomia(Categoria, Superior)

# Exercício

## Marcadores e Categorias

- 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.

A blurred office desk with a laptop and a desk lamp in front of a large window overlooking a city skyline. The scene is brightly lit, suggesting daytime. 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. The overall atmosphere is professional and focused.

# 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>)

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.

# Agrupamento

# GROUP BY

```
■ SELECT * | <campo1>[,..., <campon>]
 FROM <tabela1>[,..., <tabelan>]
 WHERE <condição/junção>
 GROUP BY <colunas_agrupar>
 HAVING <condição_grupo>
```

# Modelos de Taxi

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

## Cliente (C)

| <u>CliId</u> | Nome     | CPF            |
|--------------|----------|----------------|
| 1532         | Asdrúbal | 448.754.253-65 |
| 1755         | Doriana  | 567.387.387-44 |
| 1780         | Quincas  | 546.373.762-02 |



## Táxi (TX)



| <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   |



## Corrida (R1)

| <u>ClId</u> | <u>Placa</u> | <u>DataPedido</u> |
|-------------|--------------|-------------------|
| 1755        | DAE6534      | 15/02/2003        |
| 1982        | JDM8776      | 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 Táxis por Modelo

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

### Cliente (C)

| <u>CliId</u> | Nome     | CPF            |
|--------------|----------|----------------|
| 1532         | Asdrúbal | 448.754.253-65 |
| 1755         | Doriana  | 567.387.387-44 |
| 1780         | Quincas  | 546.373.762-02 |



### Táxi (TX)



| <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   |



### Corrida (R1)

| <u>ClId</u> | <u>Placa</u> | <u>DataPedido</u> |
|-------------|--------------|-------------------|
| 1755        | DAE6534      | 15/02/2003        |
| 1982        | JDM8776      | 18/02/2003        |

## Questão 6

# Total de Taxis por Modelo

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

### Cliente (C)

| <u>CliId</u> | Nome     | CPF            |
|--------------|----------|----------------|
| 1532         | Asdrúbal | 448.754.253-65 |
| 1755         | Doriana  | 567.387.387-44 |
| 1780         | Quincas  | 546.373.762-02 |



### Táxi (TX)



| <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   |



### Corrida (R1)

| <u>ClId</u> | <u>Placa</u> | <u>DataPedido</u> |
|-------------|--------------|-------------------|
| 1755        | DAE6534      | 15/02/2003        |
| 1982        | JDM8776      | 18/02/2003        |

# DISTINCT

## Exercício

- Modelos de taxi tomados por cada cliente (mostrar nome do 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?



# DISTINCT

## Exercício

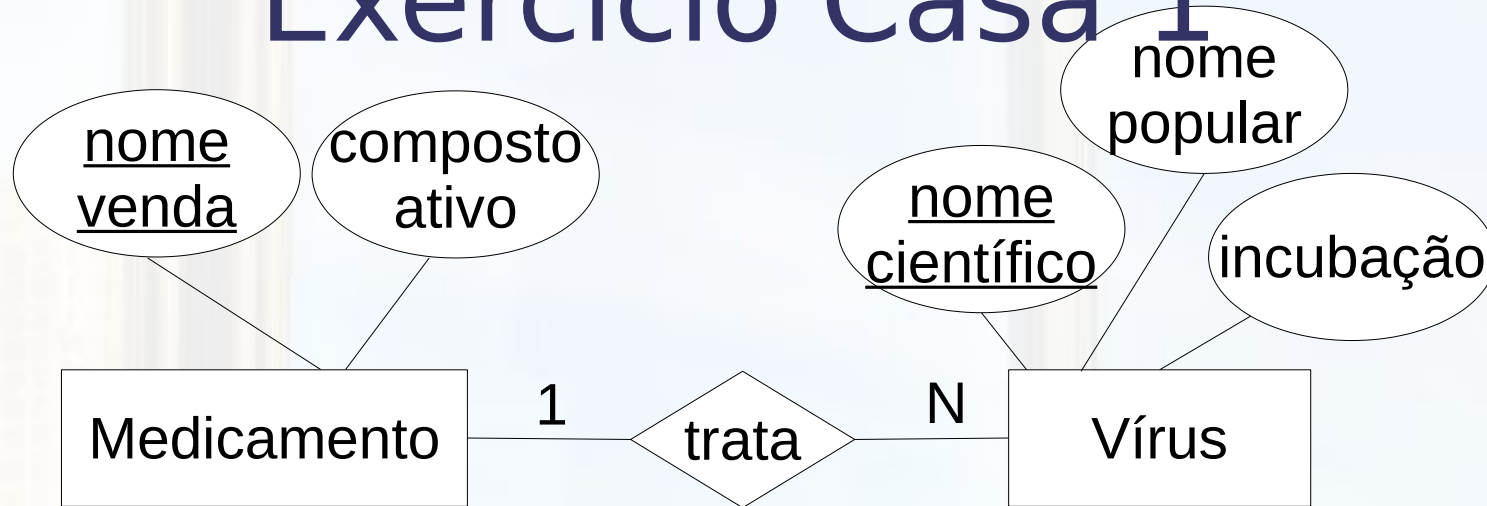
- Modelos de taxi tomados por cada cliente (mostrar nome do cliente)

```
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 Cl.Nome, T.Modelo
FROM Cliente Cl, Corrida Co, Taxi T
WHERE Cl.CliId = Co.CliId AND
 Co.Placa = T.Placa
GROUP BY Cl.Nome, T.Modelo
```

# Exercício Casa 1



medicamento(nomeVenda, compostoAtivo)

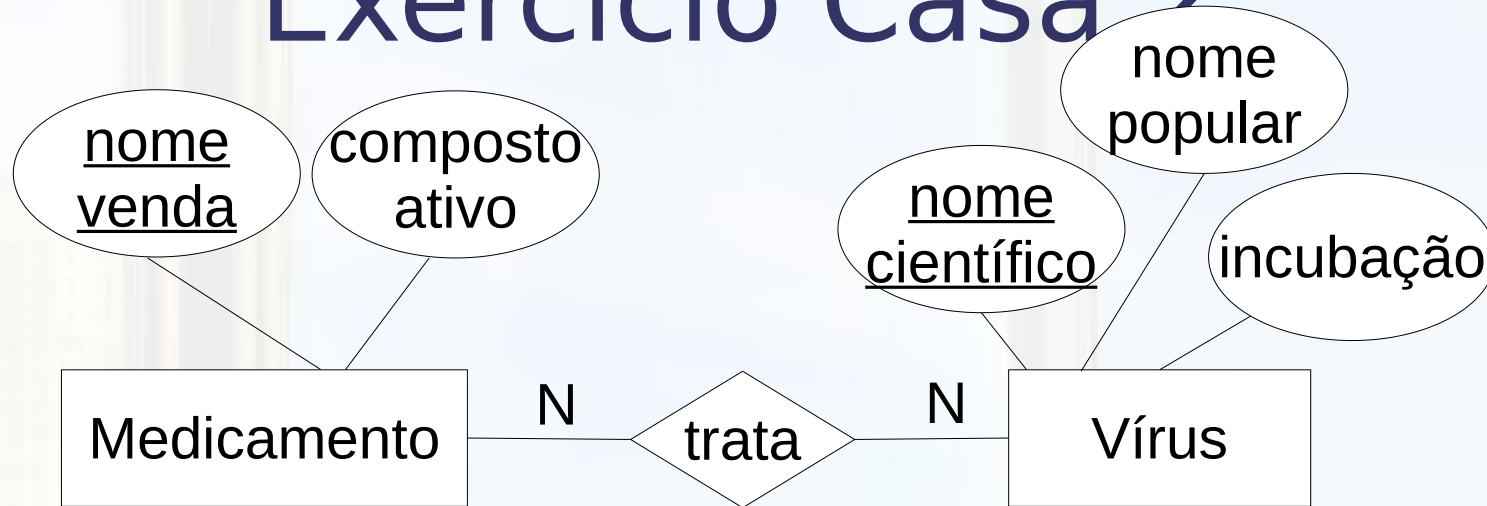
virus(nomeCientifico, nomePopular, incubacao,  
**nomeVendaMedicamento**)

- CHE: nomeVendaMedicamento para medicamento

a) Quantos vírus trata cada medicamento

b) Quantos vírus trata cada composto ativo

# Exercício Casa 2



medicamento(nomeVenda, compostoAtivo)

virus(nomeCientifico, nomePopular, incubacao)

trata(nomeVendaMedicamento, nomeCientificoVirus)

- CHE: nomeVendaMedicamento para medicamento

- CHE: nomeCientificoVirus para virus

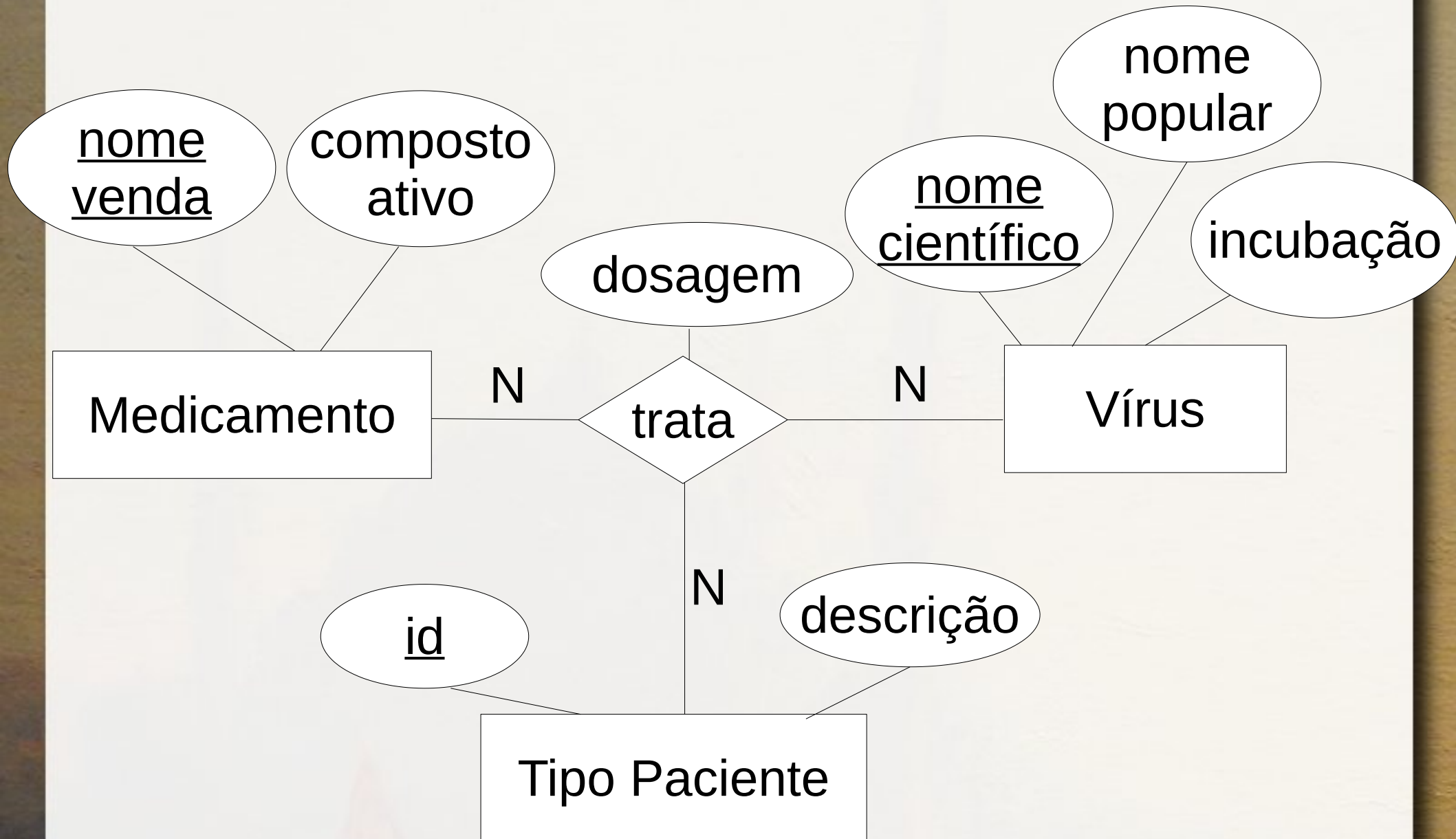
a) Quantos vírus trata cada medicamento

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 \* | <campo<sub>1</sub>>[,..., <campo<sub>n</sub>>]**  
**FROM <tabela<sub>1</sub>>[,..., <tabela<sub>n</sub>>]**  
**WHERE <condição/junção>**  
**GROUP BY <coluna\_agrupar>**  
**HAVING <condição\_grupo>**

# WHERE x HAVING

## ■ Modelos depois do ano 2000

```
SELECT Modelo, COUNT(*) NUM
FROM Taxi
WHERE AnoFab > 2000
GROUP BY Modelo;
```

## ■ Modelos com mais um taxi

```
SELECT Modelo, COUNT(*) NUM
FROM Taxi
GROUP BY Modelo
HAVING NUM > 1;
```



# Exercício

- Modelos de Táxi cujo maior Ano de Fabricação seja maior do que 2000.

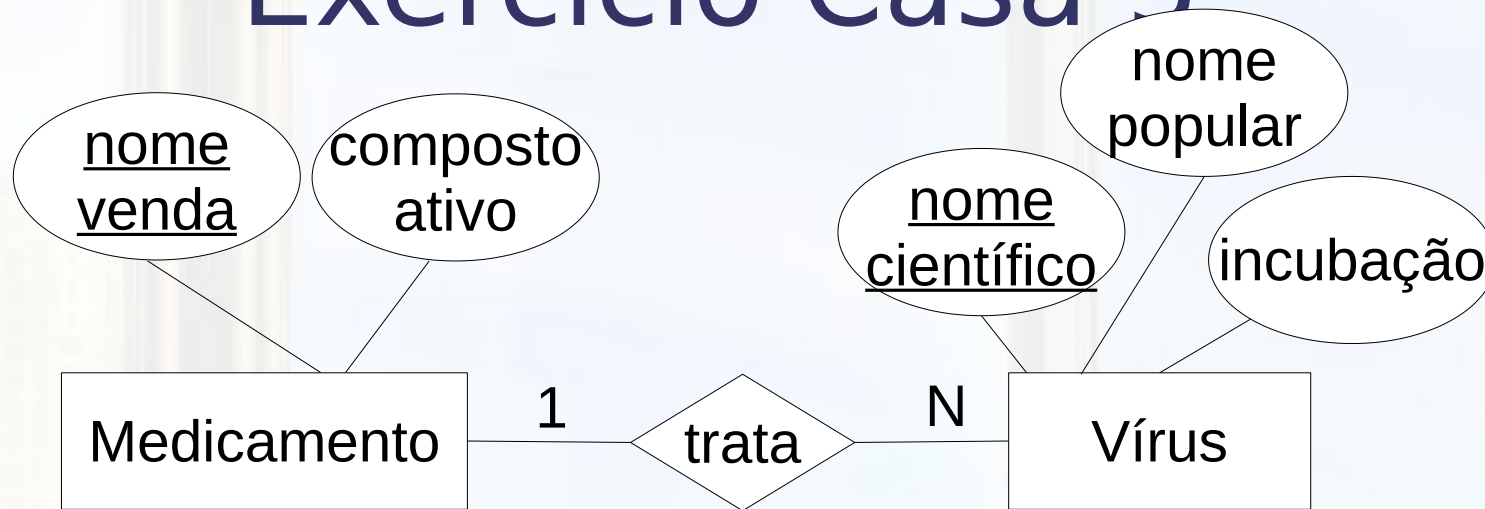
```
SELECT * | <campo1>[,..., <campon>]
FROM <tabela1>[,..., <tabelan>]
WHERE <condição/junção>
GROUP BY <coluna_agrupar>
HAVING <condição_grupo>
```

## Táxi (TX)

| <u>Placa</u>  | Marca      | Modelo  | Ano Fab |
|---------------|------------|---------|---------|
| 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  | Corso   | 1999    |



# Exercício Casa 5



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

# Visões

The image features a semi-transparent white rectangular overlay centered on a blurred background. The background is a photograph of a modern office workspace. In the foreground, a desk holds a laptop, a desk lamp with a curved neck, and various office supplies. The desk is positioned in front of a large window that provides a panoramic view of a city skyline with several tall buildings under a clear sky. The overall aesthetic is clean and professional.

# 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;
```

# Usando a View

```
■ SELECT AVG(Numero_Taxis) FROM
Contagem_Modelo;
```

# Consultas Aninhadas

The background of the slide is a blurred office scene. In the foreground, a desk holds a laptop, a desk lamp, and some papers. The desk is positioned in front of a large window that looks out onto a city with several tall buildings under a clear sky. The overall image is intentionally out of focus to create a professional and clean aesthetic.



# Tabelas de Motorista, Zona e Fila

## Motoristas e os Táxis que dirigem:

Motorista (**CNH**, Nome, CNHValid, **Placa**)

Placa → CHE Taxi

## Zonas onde Táxis fazem fila (virtualmente):

Zona (**Zona**)

## Filas virtuais de Táxis:

Fila (**Zona**, **CNH**, DataHoraIn, DataHoraOut, KmIn)

Zona → CHE Zona

CNH → CHE Motorista

# SELECT Seleção

```
SELECT * FROM Taxi WHERE AnoFab > 2000
```

| <u>Placa</u> | Marca      | Modelo  | AnoFab |
|--------------|------------|---------|--------|
| DKL4598      | Wolksvagen | Gol     | 2001   |
| DKL7878      | Ford       | Fiesta  | 2001   |
| JDM8776      | Wolksvagen | Santana | 2002   |

# SELECT IN e NOT IN

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

# IN

## ■ Zonas que receberam algum taxi na fila

### □ Sem aninhamento

```
SELECT DISTINCT F.zona
FROM Fila F;
```

### □ Com aninhamento/IN

```
SELECT Z.zona FROM Zona Z
WHERE Z.zona IN (
 SELECT DISTINCT F.zona
 FROM Fila F);
```

# Otimização do SELECT

- Zonas que receberam algum taxi na fila

```
SELECT Z.zona FROM Zona Z
 WHERE Z.zona IN (
 SELECT DISTINCT F.zona
 FROM Fila F);
```

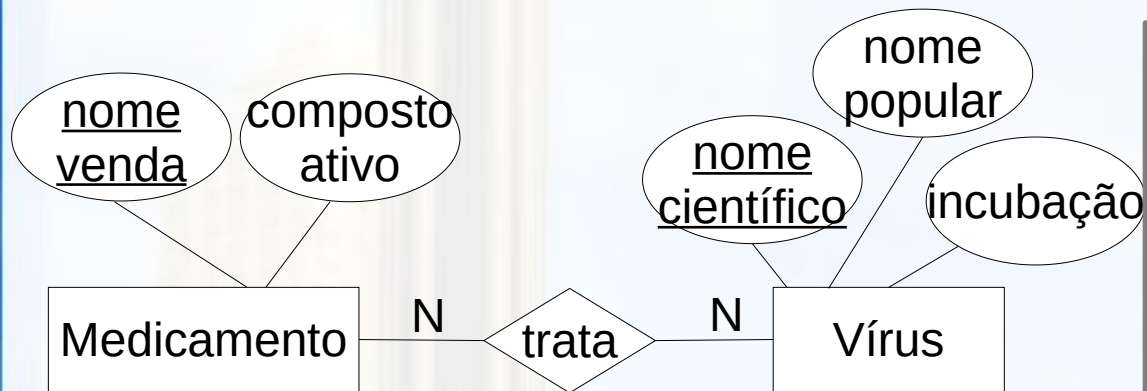
- Nesse caso, não há dependência do SELECT interno em relação ao externo
- O SELECT interno pode ser executado uma única vez e ser reusado em todas as iterações do externo

# NOT IN

- Zonas que não receberam algum taxi na fila

```
SELECT Z.zona FROM Zona Z
WHERE Z.zona NOT IN (
 SELECT DISTINCT F.zona
 FROM Fila F);
```

# Exercício



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

medicamento(nomeVenda, compostoAtivo)

virus(nomeCientifico, nomePopular, incubacao)

trata(nomeVendaMedicamento, nomeCientificoVirus)

- CHE: nomeVendaMedicamento para medicamento

- CHE: nomeCientificoVirus para virus

- Nome popular dos vírus que não são tratados pelo medicamento com composto ativo Virulex

# SELECT EXISTS e NOT EXISTS

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



# EXISTS

- Zonas que receberam algum taxi na fila

```
SELECT Z.zona FROM Zona Z
 WHERE EXISTS (
 SELECT * FROM Fila F
 WHERE F.zona = Z.zona);
```

- Nesse caso, uma variável interna `F.zona` depende de uma variável externa `Z.zona`
- O `SELECT` interno terá que ser feito para cada iteração do `SELECT` externo

# NOT EXISTS

- Zonas que não receberam algum taxi na fila

```
SELECT Z.zona FROM Zona Z
WHERE NOT EXISTS (
 SELECT * FROM Fila F
 WHERE F.zona = Z.zona);
```

# SELECT

## Comparação

- SELECT ...  
WHERE <campo> <comparação> [ALL|ANY]  
(SELECT <campo> ...)
- Modalidades de comparação:
  - Direta: apenas um valor resultante do SELECT interno
  - ALL ou ANY: mais de um valor resultante do SELECT interno
    - ALL - comparação verdadeira para todos os registros
    - ANY - comparação verdadeira para pelo menos um registro

# Comparação com um resultado

## ■ Nome dos clientes que andaram no taxi dirigido por Bonerges

```
SELECT DISTINCT Cl.nome
FROM Cliente Cl, Corrida Co
WHERE Cl.cliid = Co.cliid AND
 Co.placa = (SELECT Tx.placa
 FROM Taxi Tx, Motorista Mo
 WHERE Tx.placa = Mo.placa AND
 Mo.nome = 'Bonerges');
```

# Exercício

Quem foi o primeiro motorista a entrar em alguma fila (independentemente da fila)

Motorista (**CNH**, Nome, CNHValid, **Placa**)

Placa → CHE Taxi

Fila (**Zona**, **CNH**, DataHoraIn, DataHoraOut, KmIn)

Zona → CHE Zona

CNH → CHE Motorista

---

SELECT ...

WHERE <campo> <comparação> [ALL|ANY]

(SELECT <campo> ...)

# ALL

## ■ Quem foi o primeiro motorista a entrar na fila de cada uma das zonas

```
SELECT Mo.nome, Fi_externa.zona, Fi_externa.datahorain
FROM Motorista Mo, Fila Fi_externa
WHERE Mo.cnh = Fi_externa.cnh AND
 Fi_externa.datahorain <= ALL (
 SELECT Fi_interna.datahorain
 FROM Fila Fi_interna
 WHERE Fi_externa.zona=Fi_interna.zona);
```

# ANY

- Quem não foi o primeiro motorista a entrar na fila de cada uma das zonas

```
SELECT Mo.nome, Fi_externa.zona, Fi_externa.datahorain
FROM Motorista Mo, Fila Fi_externa
WHERE Mo.cnh = Fi_externa.cnh AND
 Fi_externa.datahorain > ANY (
 SELECT Fi_interna.datahorain
 FROM Fila Fi_interna
 WHERE Fi_externa.zona=Fi_interna.zona)
ORDER BY Fi_externa.zona;
```

# Exercício para Casa 6

- 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 como Tabela Aninhada

```
SELECT ...
FROM
 (SELECT ...) <alias>
```

- O resultado do SELECT no FROM se comporta como uma nova tabela.
- Se for dado um <alias> esse será o nome da tabela.

# SELECT como Tabela Aninhada

## ■ Média de Táxis por Zona

```
SELECT AVG(FT.n_taxis)
 FROM (SELECT F.zona, COUNT(*) n_taxis
 FROM Fila F
 GROUP BY F.zona) FT
```



# **SELECT Aninhado em UPDATE e DELETE**

# Aninhando UPDATE/SELECT

- Invalidando o CNH de todos os motoristas que não aparecem em nenhuma fila.

```
UPDATE Motorista M
 SET M.CNHValid = 0
 WHERE M.CNH NOT IN (
 SELECT DISTINCT F.CNH
 FROM Fila F);
```

# Aninhando DELETE/SELECT

- Excluindo os motoristas que não aparecem em nenhuma fila.

```
DELETE FROM Motorista M
WHERE M.CNH NOT IN (
 SELECT DISTINCT F.CNH
 FROM Fila F);
```

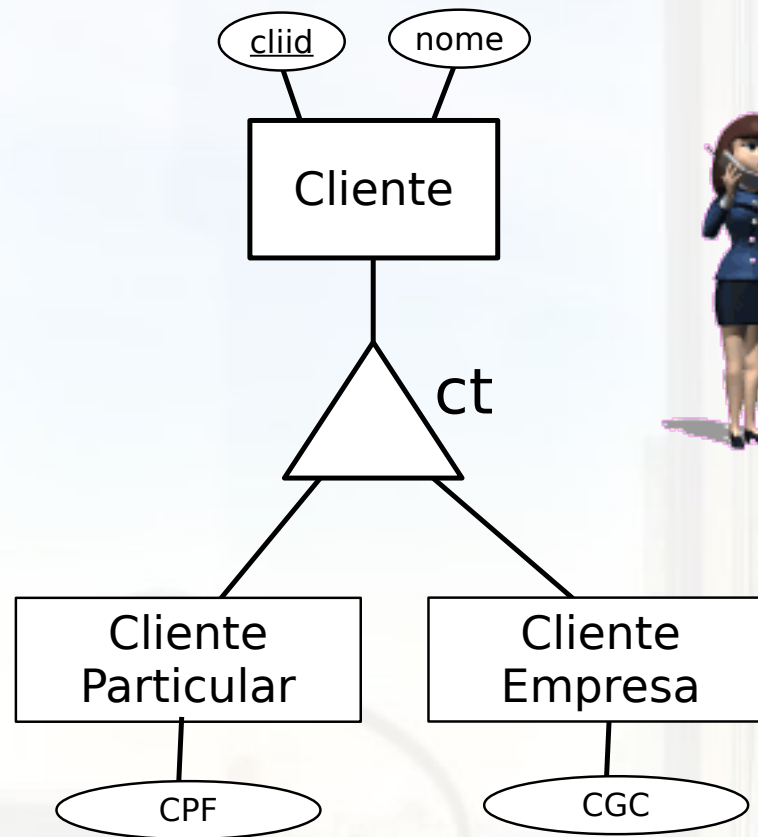
# Operações de Conjunto

The background of the slide is a blurred photograph of a modern office. In the foreground, a desk is visible with a laptop, a desk lamp, and some papers. 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.

# União, Interseção e Diferença

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

# Esquema Conceitual Exemplo Táxis



Cliente Particular - indivíduo que irá pagar a conta

Cliente Empresa - funcionário de uma empresa conveniada, para a qual a conta será enviada



# 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 |

# Mesmo Esquema

- As operações de conjunto exigem que as tabelas tenham o mesmo esquema.
- Pode ser feito através de uma projeção:

```
SELECT cliid, nome
FROM Cliente;
```

```
SELECT cliid, nome
FROM ClienteEmpresa;
```

# UNION

- Todos os clientes das duas tabelas (Cliente e ClienteEmpresa)

```
SELECT cliid, nome
FROM Cliente
```

## **UNION**

```
SELECT cliid, nome
FROM ClienteEmpresa;
```

# INTERSECT

- Clientes que aparecem em ambas as tabelas.

```
SELECT cliid, nome
FROM Cliente
```

## **INTERSECT**

```
SELECT cliid, nome
FROM ClienteEmpresa;
```

# EXCEPT

- Pessoas que estão na tabela Cliente e que não estão na tabela ClienteEmpresa.

```
SELECT cliid, nome
FROM Cliente
```

## **EXCEPT**

```
SELECT cliid, nome
FROM ClienteEmpresa;
```

# Combinando Operações

- Clientes que estão exclusivamente em uma das tabelas (não aparecem em ambas as tabelas).

```
(SELECT cliid, nome
FROM Cliente
```

## **UNION**

```
SELECT cliid, nome
FROM ClienteEmpresa)
```

## **EXCEPT**

```
(SELECT cliid, nome
FROM Cliente
```

## **INTERSECT**

```
SELECT cliid, nome
FROM ClienteEmpresa)
```

# Agradecimentos

- Luiz Celso Gomes Jr (professor desta disciplina em 2014) pela contribuição na disciplina e nos slides.  
Página do Celso:  
<http://dainf.ct.utfpr.edu.br/~gomesjr/>
- Patrícia Cavoto (professora desta disciplina em 2016) pela contribuição na disciplina e nos slides.  
Página da Patrícia: <http://patricia.cavoto.com.br>

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/>