



UNIPAC - CENTRO UNIVERSITÁRIO PRESIDENTE ANTÔNIO CARLOS  
CAMPUS BARBACENA

Bacharelado em Ciência da Computação



# Banco de Dados

## Material de Apoio

**Parte VII – Modelo Físico de Dados**

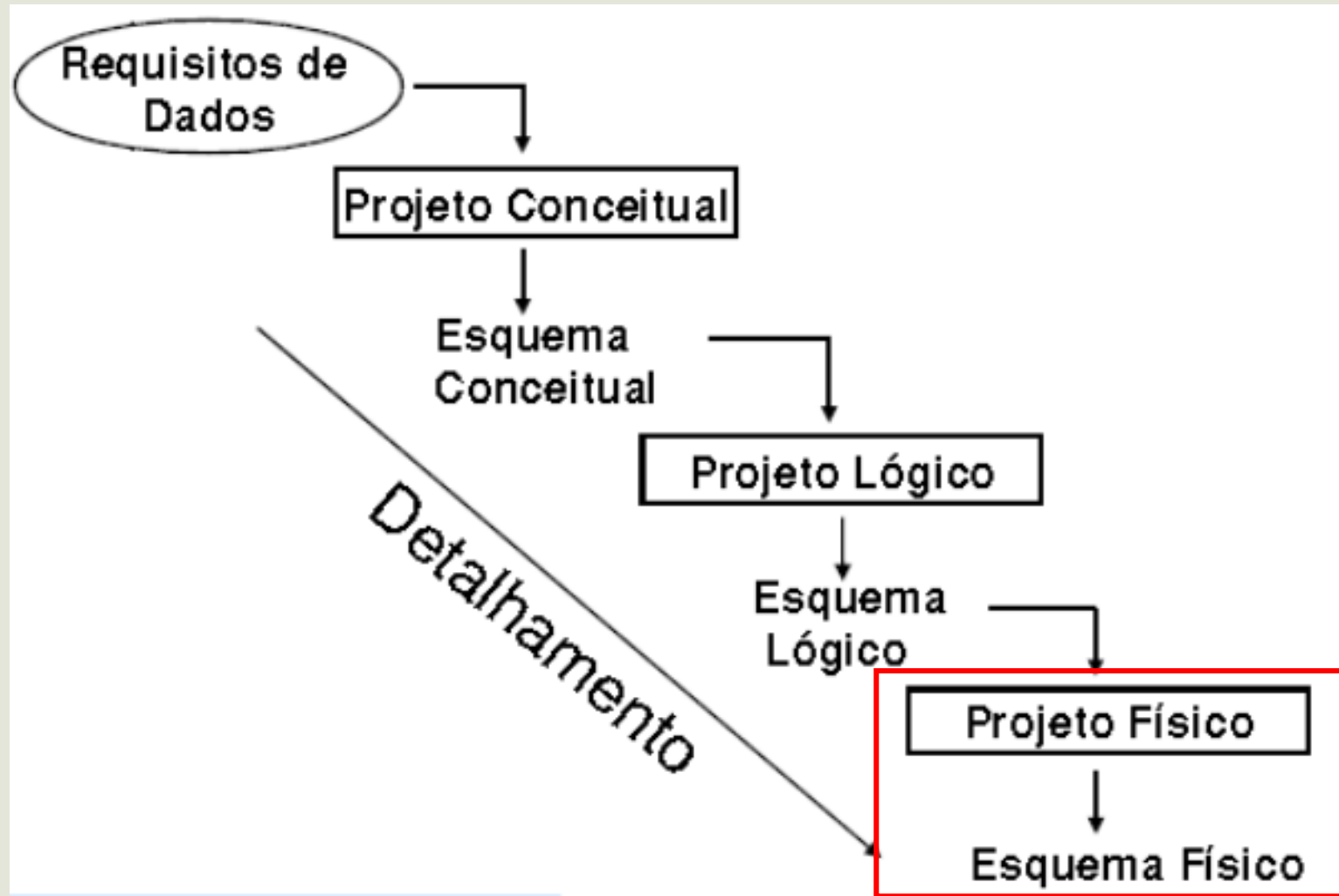
Prof. José Osvano da Silva, PMP, PSM I  
joseosvano@unipac.br

1º sem / 2022

# Sumário

- Introdução
- Linguagens de Query
- SQL
- Caso Prático – Taxis
- Exercícios

# Introdução



# Linguagens de Query

- Para manipulação e recuperação de dados
- Linguagens de Query (LQ) em BD:
  - Fundamentação formal
  - Subsidiária otimização
- LQ  $\neq$  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 Query

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

- 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

- **SQL** – Structured Query Language
  - Data Definition Language (DDL)
    - Instruções para definição de dados
  - Data Manipulation Language (DML)
    - Instruções para consultas e atualizações
  - Data Control Language (DCL)
    - Instruções para controle de dados
  - Linguagem de Transação de Dados (DTL)
    - Commit, Rollback
  - Linguagem de Consulta de Dados (DQL)
    - Select

# 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

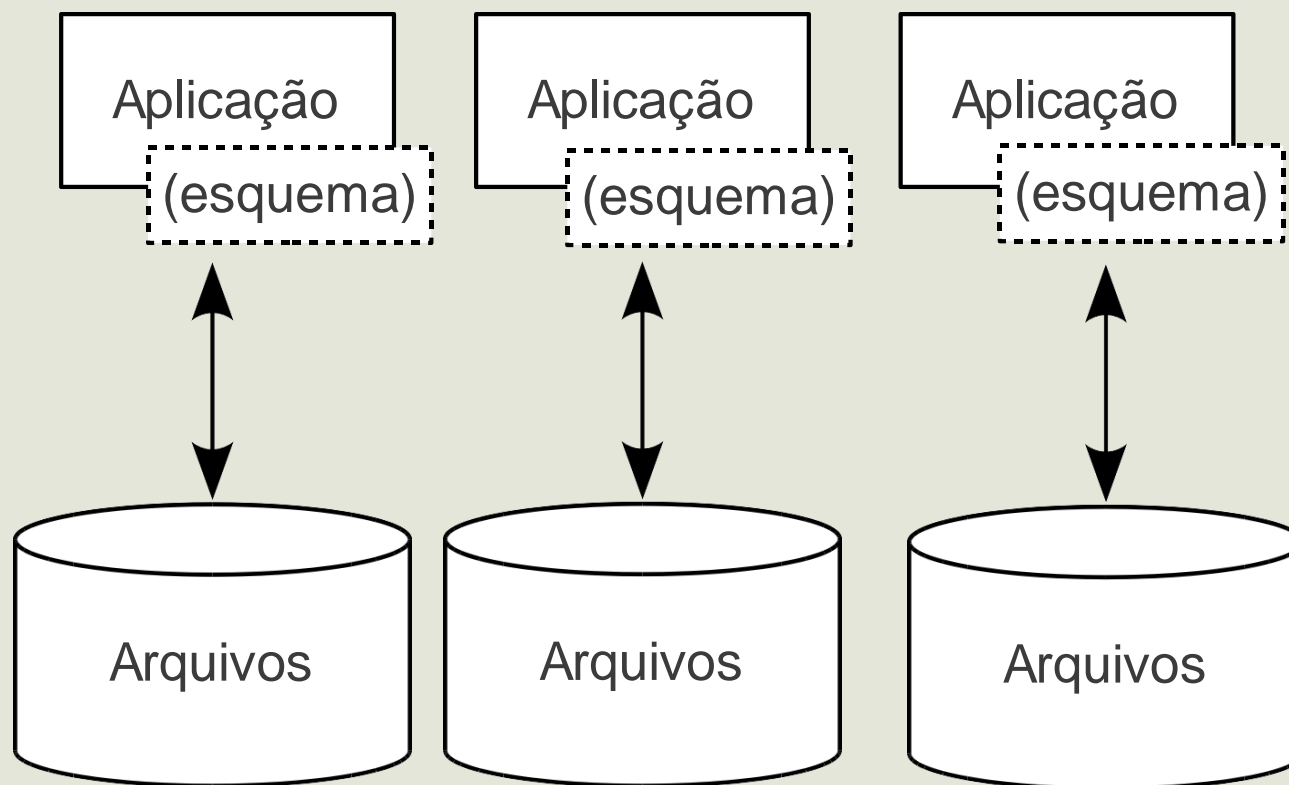


# SQL - Padronização

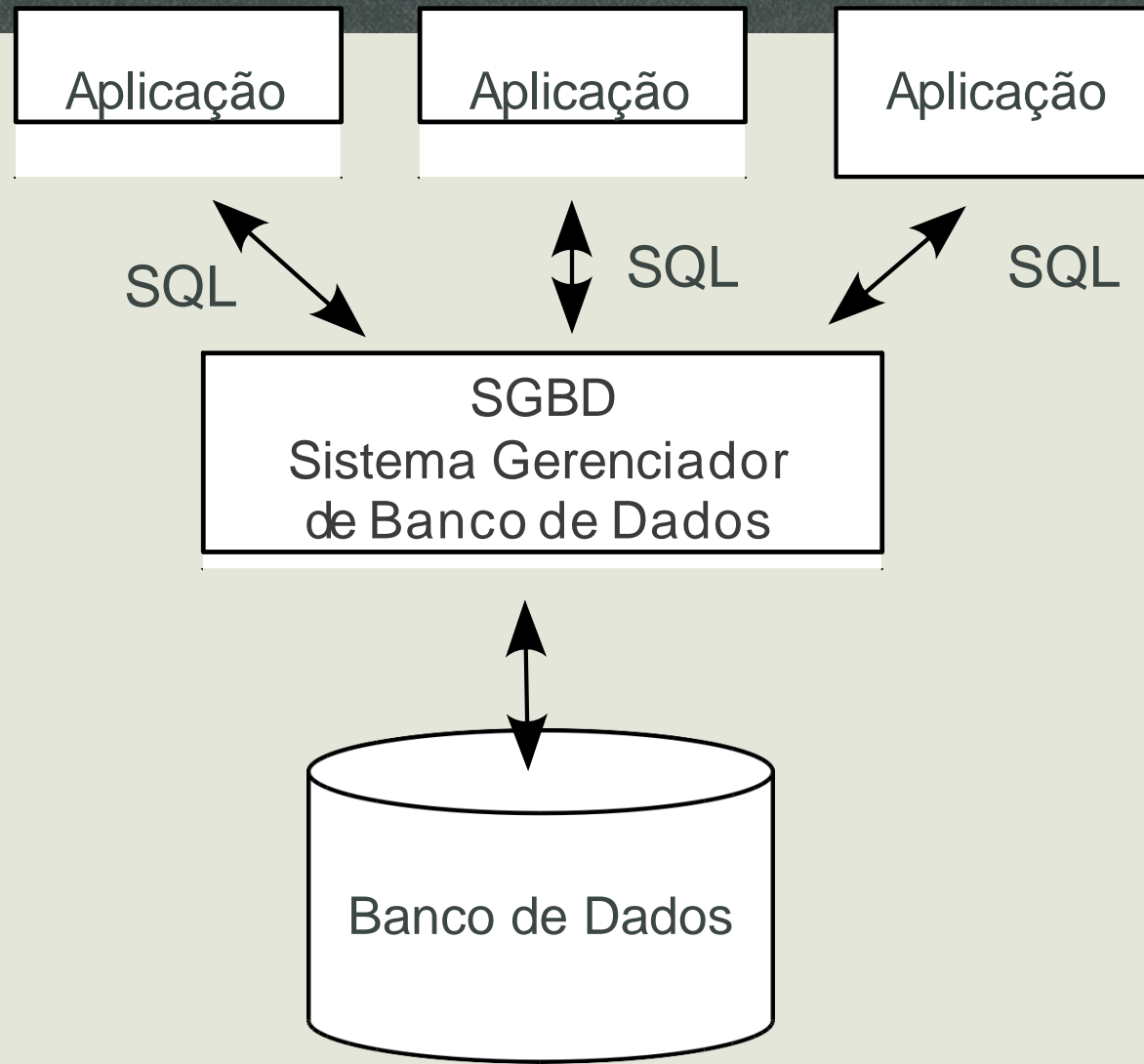
- ANSI + ISO
- SQL-86 ou SQL1
- SQL-92 ou SQL2
- SQL:1999 ou SQL3
- SQL:2003
- **SQL:2006 → adotaremos essa versão**
- SQL:2008
- SQL:2011
- SQL:2016

# Aplicações e Armazenamento

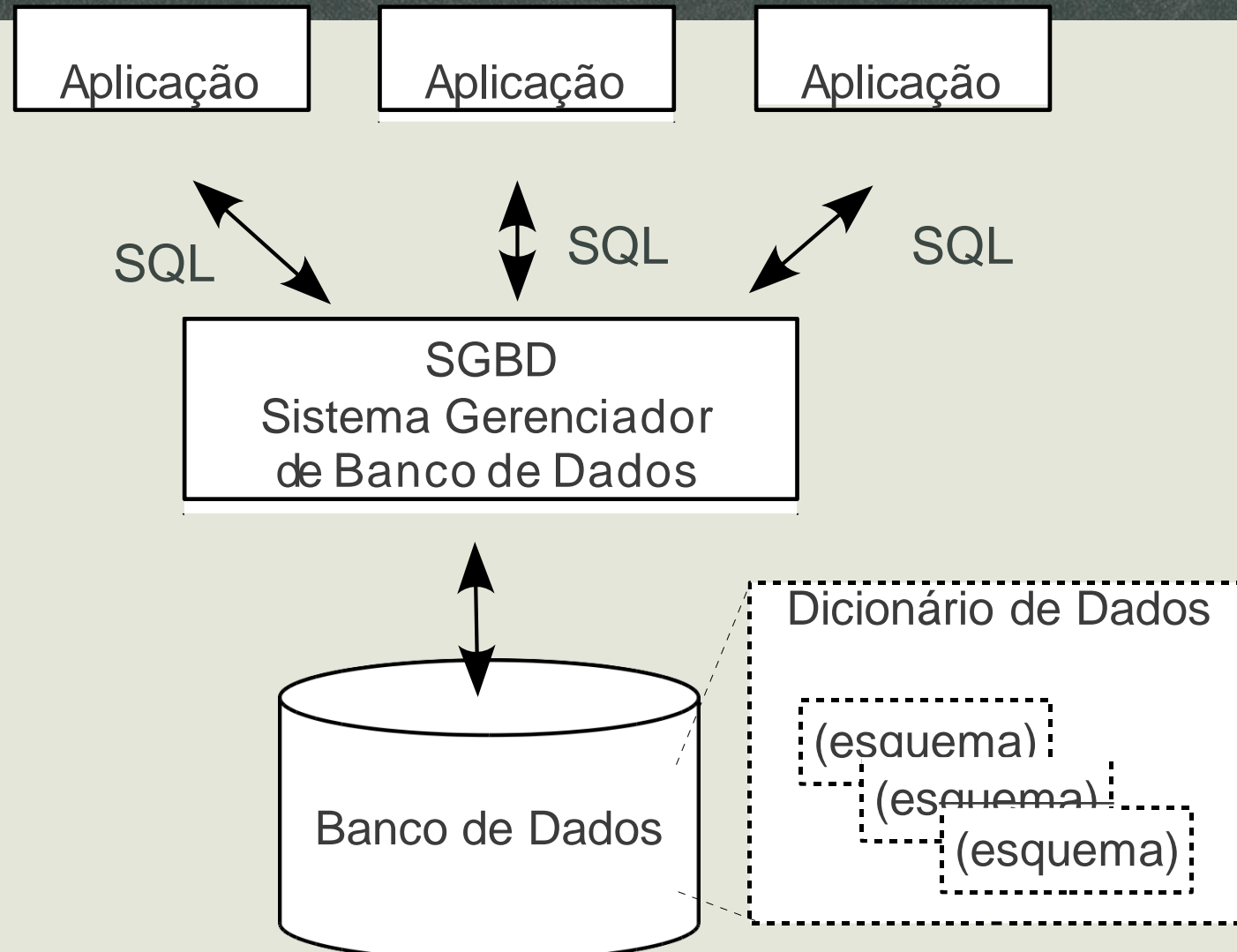
## Arquivos



# Aplicações e Armazenamento SGBD



# Dicionário de Dados



# Caso Prático - Taxis

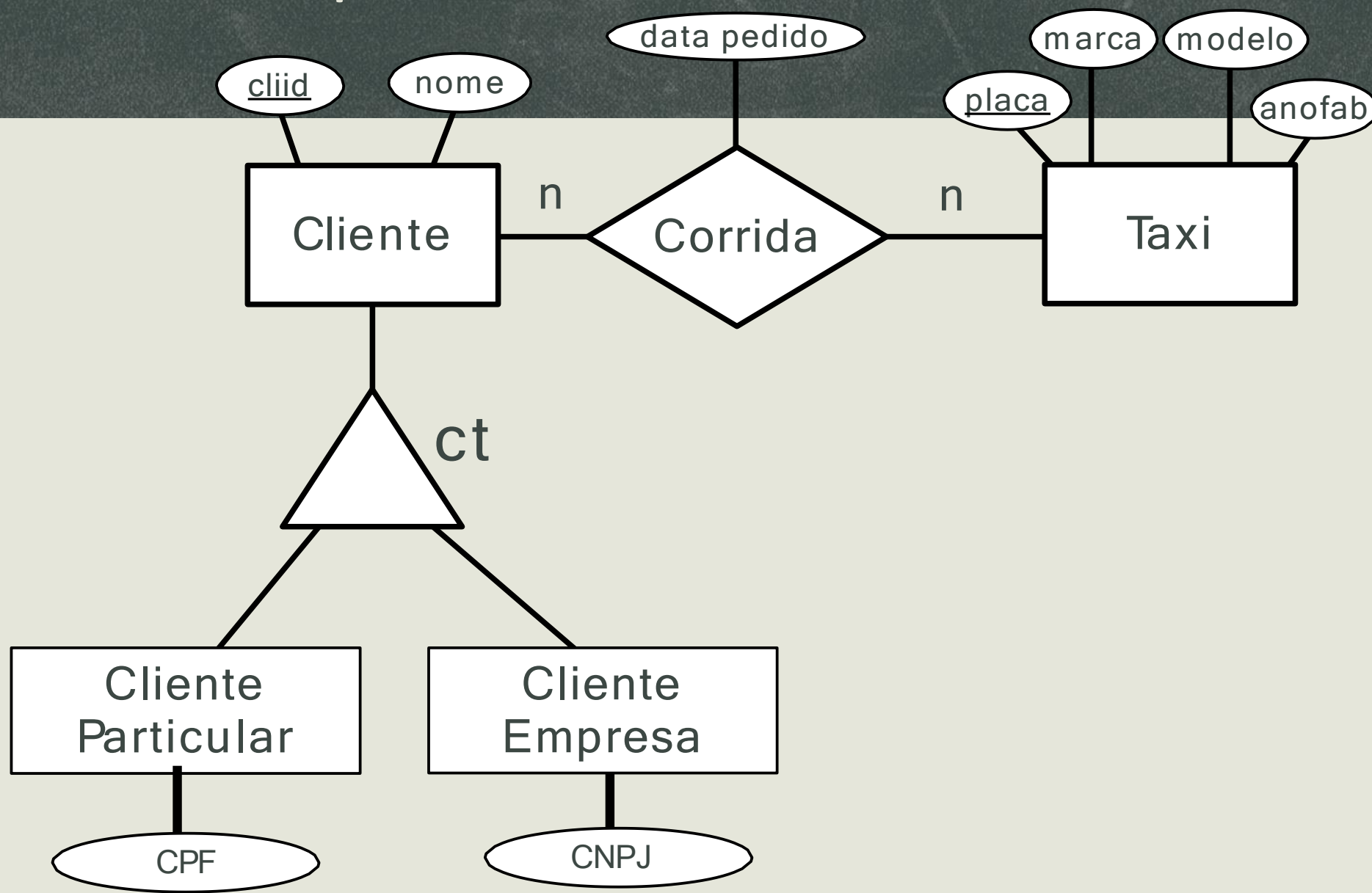


# Esquema Conceitual

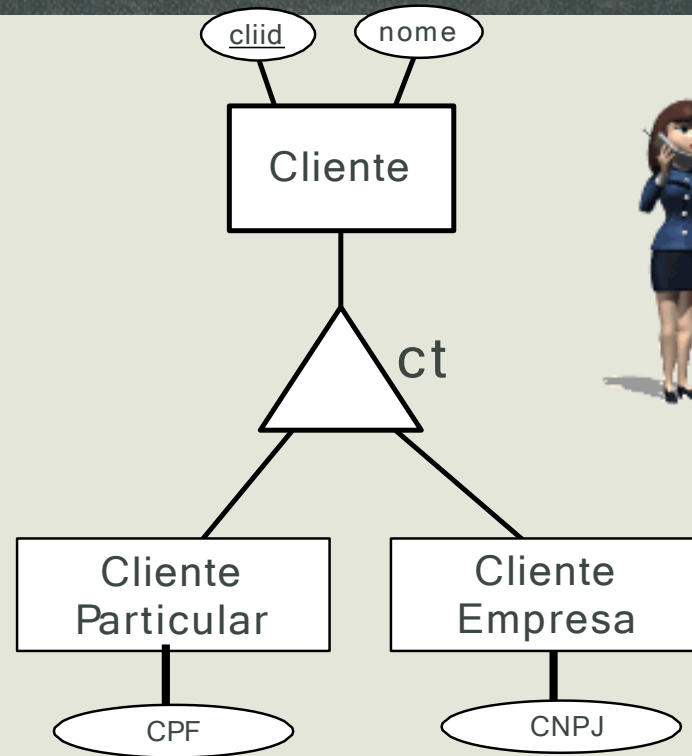
## Exemplo Táxis

- 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 que vamos adotar.

# 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 li Id</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 li Id</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>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	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



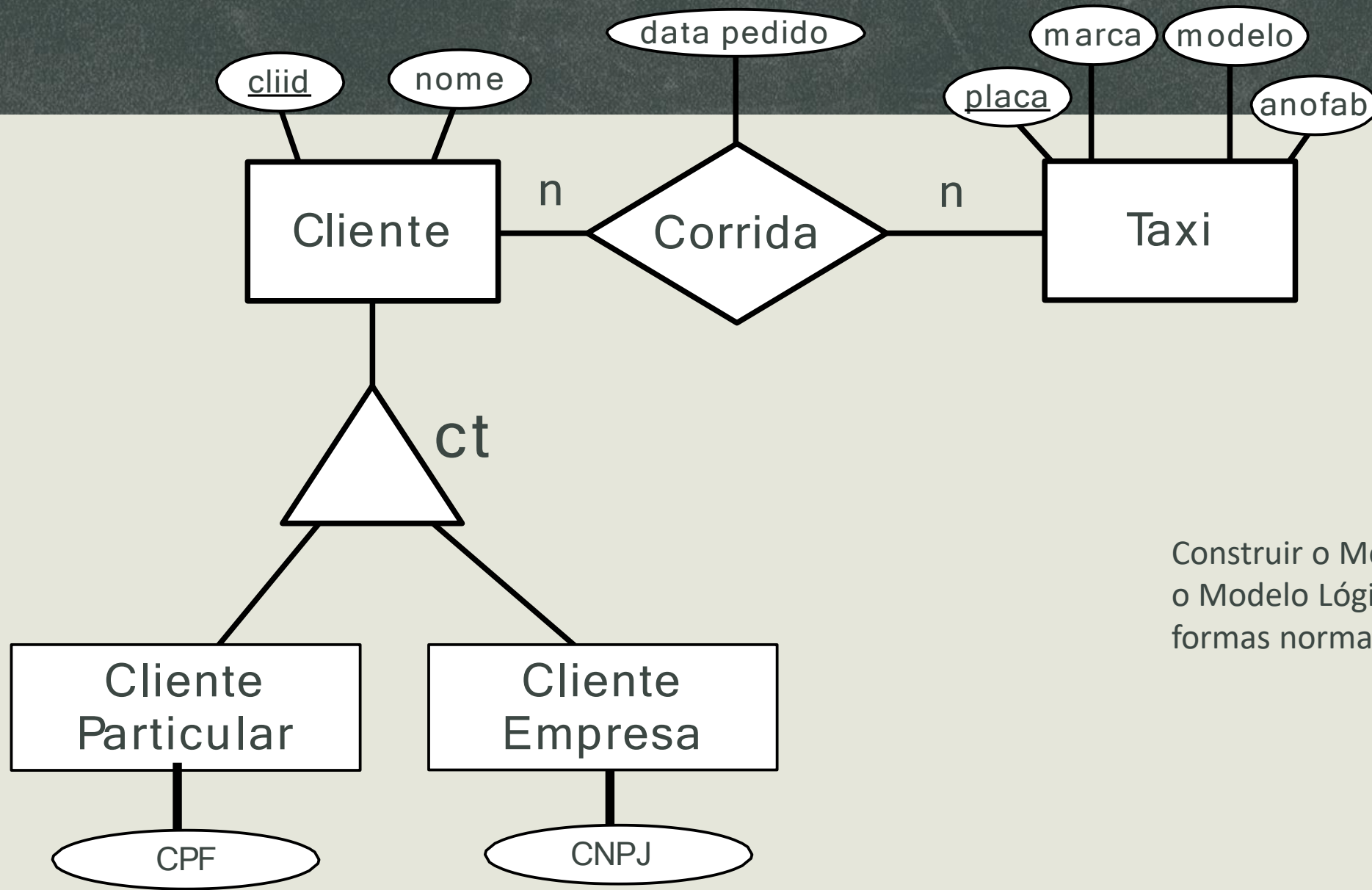
## Corrida (R1)

<u>C l i d</u>	<u>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





# Exercício de Fixação 08



Construir o Modelo Relacional,  
o Modelo Lógico e validar as  
formas normais.



# DDL - Data Definition Language

# Definições em SQL

- Terminologia:
  - **Tabela, linha e coluna** usados para os termos do modelo relacional *relação, tupla e atributo*
- Instruções CREATE
  - Principal comando SQL para a definição de dados

# Conceitos de esquema e catálogo em SQL

- **Esquema SQL**
  - Identificado por um **nome de esquema**
  - Inclui um **identificador de autorização e descritores** para cada elemento
- **Instrução CREATE SCHEMA**
  - `CREATE SCHEMA EMPRESA;`
- **Catálogo**
  - Coleção nomeada de esquemas em um ambiente SQL
  - Contém um esquema especial, chamado `INFORMATION_SCHEMA`



## O comando CREATE TABLE em SQL


- Especificar uma nova relação
  - Dar um nome
  - Especificar atributos e restrições iniciais, depois restrições de chave e integridade referencial.
- Pode especificar o esquema:
  - CREATE           TABLE           EMPRESA.FUNCIONARIO           ...
  - ou
  - CREATE TABLE FUNCIONARIO ...



## O comando CREATE TABLE em SQL (cont.)

```
CREATE TABLE <nome_da_tabela> (  
     $C_1$   $D_1$  [(restrição atributo)],  
     $C_2$   $D_2$  [(restrição atributo)],  
     $C_n$   $D_n$  [(restrição atributo)],  
PRIMARY KEY <lista_de_Colunas>,  
FOREIGN KEY <nome_da_coluna> REFERENCES  
    <nome_tab_ref>(<nome_da_coluna_ref>));
```

restrições iniciais



cada  $C_i$  é uma coluna no esquema da tabela  
 $D_i$  é o tipo de dado no domínio da coluna  $C_i$

## O comando CREATE TABLE em SQL (cont.)

- Algumas chaves estrangeiras podem causar erros
  - Especificadas por:
    - Referências circulares
    - Ou porque dizem respeito a uma tabela que ainda não foi criada

## O comando CREATE TABLE em SQL (cont.)

```
CREATE TABLE FUNCIONARIO
(Pnome          VARCHAR(15)          NOT NULL,
Minicial        CHAR,
Unome          VARCHAR(15)          NOT NULL,
Cpf             CHAR(11),            NOT NULL,
Datanasc       DATE,
Endereço       VARCHAR(30),
Sexo           CHAR,
Salario        DECIMAL(10,2),
Cpf_supervisor CHAR(11),            NOT NULL,
Dnr            INT
PRIMARY KEY (Cpf),
FOREIGN KEY (Cpf_supervisor) REFERENCES FUNCIONARIO(Cpf),
FOREIGN KEY (Dnr) REFERENCES DEPARTAMENTO(Dnumero) );

CREATE TABLE DEPARTAMENTO
( Dnome          VARCHAR(15)          NOT NULL,
Dnumero         INT                  NOT NULL,
Cpf_gerente     CHAR(11),            NOT NULL,
Data_inicio_gerente DATE,
PRIMARY KEY (Dnumero),
UNIQUE (Dnome),
FOREIGN KEY (Cpf_gerente) REFERENCES FUNCIONARIO(Cpf) );
```

# O comando ALTER TABLE em SQL

```
ALTER TABLE nome_tabela
ADD [COLUMN] nome_atributo_1 tipo_1 [{RIs}]
    [{, nome_atributo_n tipo_n [{RIs}]]]
|
* MODIFY [COLUMN] nome_atributo_1 tipo_1 [{RIs}]
    [{, nome_atributo_n tipo_n [{RIs}]]]
|
DROP COLUMN nome_atributo_1
    [{, nome_atributo_n }]
|
ADD CONSTRAINT nome_RI_1 def_RI_1
    [{, nome_RI_n def_RI_n}]
|
DROP CONSTRAINT nome_RI_1
    [{, nome_RI_n}]
|
[ADD|DROP] [PRIMARY KEY ...|FOREIGN KEY ...]
```

\* Não é aceito em todos os SGBDs



## O comando DROP TABLE em SQL

- Exclui uma relação
- Pode excluir a relação:
  - DROP TABLE EMPRESA.FUNCIONARIO ...  
ou
  - DROP TABLE FUNCIONARIO ...



# Tipos de dados de atributo em SQL

- **Tipos de dados básicos**

- **Tipos de dados numérico**

- Incluem números inteiros: INTEGER ou INT e SMALLINT
    - Números de ponto flutuante (reais): FLOAT ou REAL e DOUBLE PRECISION
    - Números de ponto flutuante empacotado: DECIMAL(i,j) ou DEC(i,j) ou NUMERIC(i,j)
      - Onde i é a precisão e j, a escala.

- **Tipos de dados de cadeia de caracteres**

- Tamanho fixo: CHAR(n) ou CHARACTER(n)
    - Tamanho variável: VARCHAR(n) ou CHAR VARYING(n) ou CHARACTER VARYING(n)

## Tipos de dados de atributo em SQL (cont.)

- Tipos de dados de **cadeia de bits**
  - Tamanho fixo: BIT( $n$ )
  - Tamanho variável: BIT VARYING( $n$ )
- Tipo de dado **booleano**
  - Valores TRUE ou FALSE ou NULL
- Tipo de dados **DATE**
  - Componentes são DAY, MONTH e YEAR na forma DD-MM-YYYY.
  - No MySQL a forma é YYYY-MM-DD.
- Tipo de dados **TIME**
  - Componentes são HOUR, MINUTE e SECOND na forma HH:MM:SS

## Tipos de dados de atributo em SQL (cont.)

- Tipos de dados adicionais
  - Tipo de dado **Timestamp** (TIMESTAMP)
    - Inclui os campos DATE e TIME
    - Mais um mínimo de seis posições para frações decimais de segundos
    - Qualificador opcional WITH TIME ZONE
  - Tipo de dado INTERVAL
    - Especifica valor relativo que pode ser usado para incrementar ou decrementar um valor absoluto de uma data, hora ou timestamp

# Domínios em SQL

- Domínio
  - Nome usado para especificar o tipo de dado de um atributo que é usado várias vezes num esquema.
  - É mais fácil mudar o tipo de dado de um domínio do que diversos atributos.
  - Melhora a legibilidade do esquema.
  - Exemplo:
    - **CREATE DOMAIN TIPO\_CPF AS CHAR(11);**

# Especificando restrições em SQL

- Restrições básicas:
  - Restrições de chave e integridade referencial
  - Restrições sobre domínios de atributo e NULLs
  - Restrições sobre tuplas individuais dentro de uma relação



## Especificando restrições de atributo e defaults de atributo

- **NOT NULL**
  - Valor NULL não é permitido para um determinado atributo.
  - Ex: ProjNome VARCHAR(15) **NOT NULL**.
- Valor padrão
  - **DEFAULT** <valor>
  - Usado para inicializar o valor de um atributo.
  - Ex: Estado CHAR(2) **DEFAULT** 'MG'.
  - Ex: Salario FLOAT **DEFAULT** 0 **NOT NULL**;
- cláusula **CHECK**
- Dnumero INT **NOT NULL CHECK** (Dnumero > 0 AND Dnumero < 21);

# Especificando restrições de chave e integridade referencial

- Cláusula **PRIMARY KEY**
  - Especifica um ou mais atributos que compõem a chave primária de uma relação
  - Dnumero INT PRIMARY KEY;
- Cláusula **UNIQUE**
  - Especifica chaves alternativas (secundárias)
  - Dnome VARCHAR(15) UNIQUE;

```
CREATE TABLE DEPARTAMENTO
(Dnome VARCHAR(15) NOT NULL,
Dnumero INT,
PRIMARY KEY (Dnumero),
UNIQUE (Dnome))
```

## Especificando restrições de chave e integridade referencial (cont.)

- Cláusula **FOREIGN KEY**
  - Opção **RESTRICT**: rejeita atualização sobre violação
  - Conectado à cláusula de **ação de disparo referencial**
    - Opções incluem:
      - **SET NULL**
      - **SET DEFAULT**
      - **CASCADE**

## Especificando restrições de atributo e defaults de atributo

```
CREATE TABLE FUNCIONARIO
( ...,
  Dnr          INT          NOT NULL   DEFAULT 1,
  CONSTRAINT CHPFUNC
    PRIMARY KEY (Cpf),
  CONSTRAINT CHESUPERFUNC
    FOREIGN KEY (Cpf_supervisor) REFERENCES FUNCIONARIO(Cpf)
    ON DELETE SET NULL   ON UPDATE CASCADE,
  CONSTRAINT CHEDEPFUNC
    FOREIGN KEY (Dnr) REFERENCES DEPARTAMENTO(Dnumero)
    ON DELETE SET DEFAULT ON UPDATE CASCADE);
CREATE TABLE DEPARTAMENTO
( ...,
  Cpf_gerente  CHAR(11)     NOT NULL   DEFAULT '88866555576',
  ...,
  CONSTRAINT CHPDEP
    PRIMARY KEY (Dnumero),
  CONSTRAINT CHSDEP
    UNIQUE (Dnome),
  CONSTRAINT CHEGERDEP
    FOREIGN KEY (Cpf_gerente) REFERENCES FUNCIONARIO(Cpf)
    ON DELETE SET DEFAULT ON UPDATE CASCADE);
CREATE TABLE LOCALIZACAO_DEP
( ...,
  PRIMARY KEY (Dnumero, Dlocal),
  FOREIGN KEY (Dnumero) REFERENCES DEPARTAMENTO(Dnumero)
    ON DELETE CASCADE   ON UPDATE CASCADE);
```

**Figura 4.2**

Exemplo ilustrando como os valores de atributo default e as ações disparadas por integridade referencial são especificadas em SQL.

## Especificando restrições sobre tuplas usando CHECK

- Cláusula CHECK ao final de uma instrução CREATE TABLE
  - Aplicam a cada tupla individualmente
  - CHECK (Dep\_data\_criacao <= Data\_inicio\_gerente);



# Resumo

- Comandos para definição de esquemas

- **create table**

- define a estrutura da tabela, suas restrições de integridade e cria uma tabela vazia

- **alter table**

- modifica a definição de uma tabela (I / E / A atributos; I / E RIs)
    - RIs básicas:
      - atributos chave não podem ser removidos de uma tabela
      - atributos NOT NULL não podem ser inseridos em uma tabela

- **drop table**

- remove uma tabela com todas as suas tuplas

## Exercício de Fixação 09

- Criar o Modelo Físico para o projeto das corridas de Taxi.
- Enviar para a avaliação.

# Dúvidas



**José Osvano da Silva**  
[joseosvano@unipac.br](mailto:joseosvano@unipac.br)