

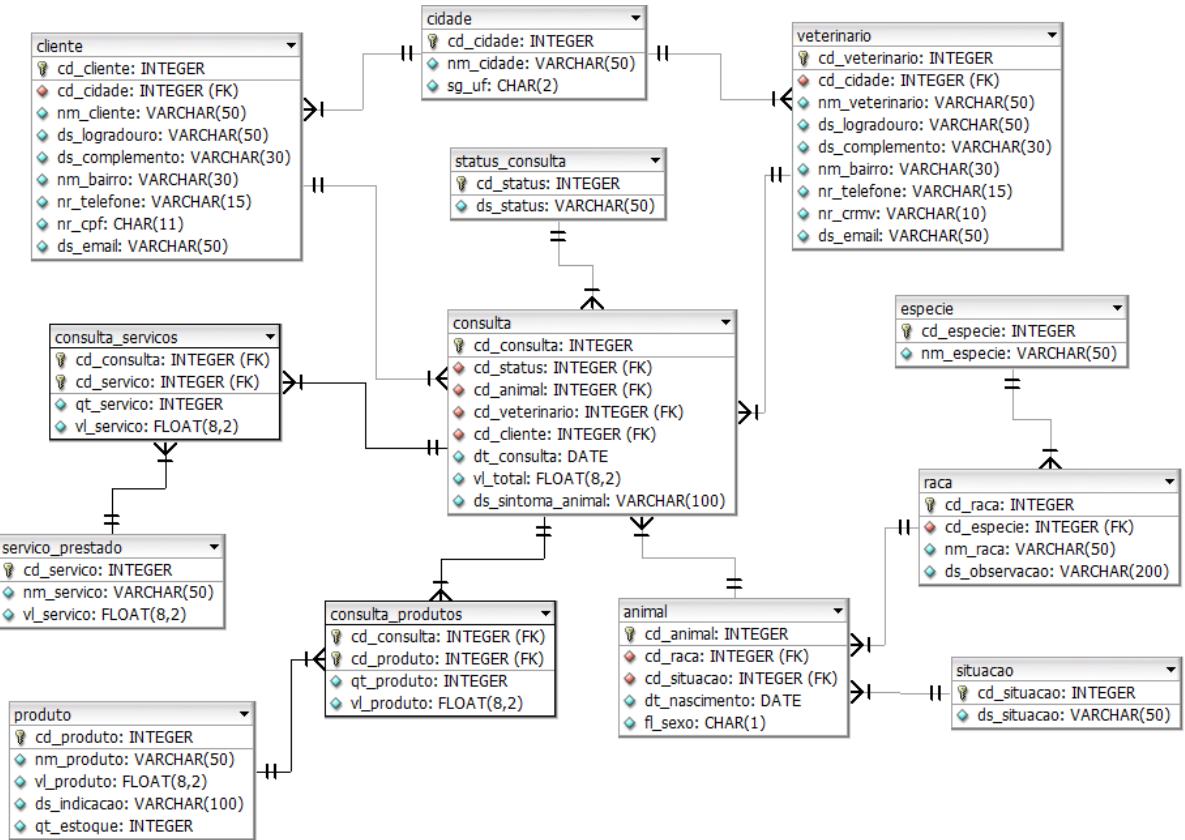
# **Modelagem**

**Professor: Henrique Delegrego**

# Modelagem

## Modelo entidade-relacionamento (MER)

- Abordagem conceitual do banco de dados
- Representa visualmente as entidades, seus atributos e os relacionamentos entre eles
- Para determinar se uma situação deve ser uma entidade:
  - Há necessidade de guardar informações sobre cada objeto?
  - Há mais de um objeto desse tipo?
  - É possível identificar cada objeto unicamente?
- Diante das respostas **positivas** obtidas, podemos afirmar que essa situação é de fato uma entidade



# Modelagem

## Tabela

- Em um banco de dados, uma tabela é uma estrutura de objetos usada para guardar e organizar informações
- Armazena dados em linhas e colunas
- Cada coluna possui um nome (ex.: cd\_aluno, nm\_aluno, dt\_nascimento)
- Cada coluna tem um tipo de dado específico (ex.: VARCHAR, INTEGER, DATE)

| Aluno          |              |
|----------------|--------------|
| cd_aluno:      | INTEGER      |
| cd_curso:      | INTEGER (FK) |
| nm_aluno:      | VARCHAR(50)  |
| ds_email:      | VARCHAR(50)  |
| dt_nascimento: | DATE         |

# Modelagem

## Chaves Primárias e Estrangeiras

- **Primary Key (PK)**
  - Identificador único e obrigatório da tabela
  - Imutável
  - Não nulo
- **Foreign Key (FK)**
  - Referência da PK em outra tabela

| Aluno |                        |
|-------|------------------------|
| !     | cd_aluno: INTEGER      |
| ◆     | cd_curso: INTEGER (FK) |
| ◆     | nm_aluno: VARCHAR(50)  |
| ◆     | ds_email: VARCHAR(50)  |
| ◆     | dt_nascimento: DATE    |

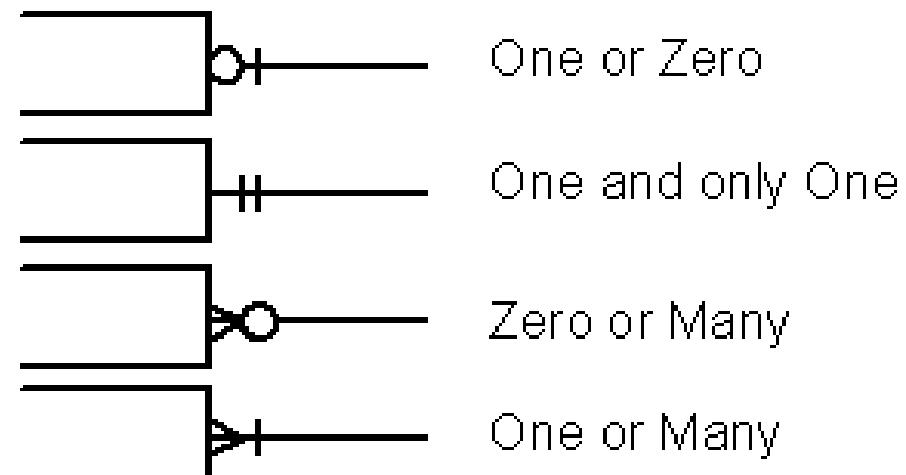
| Curso |                       |
|-------|-----------------------|
| !     | cd_curso: INTEGER     |
| ◆     | nm_curso: VARCHAR(50) |
| ◆     | nr_horas: INTEGER     |

# Modelagem

## Relacionamento

- Os relacionamentos de banco de dados são associações entre tabelas que são criadas usando instruções de junção para recuperar dados
- São elas:
  - Um para um
  - Uma para muitos
  - Muitos para muitos
- Usamos verbos para fazer a relação entre tabelas
- É usada a notação pé de galinha (**crow's foot**)

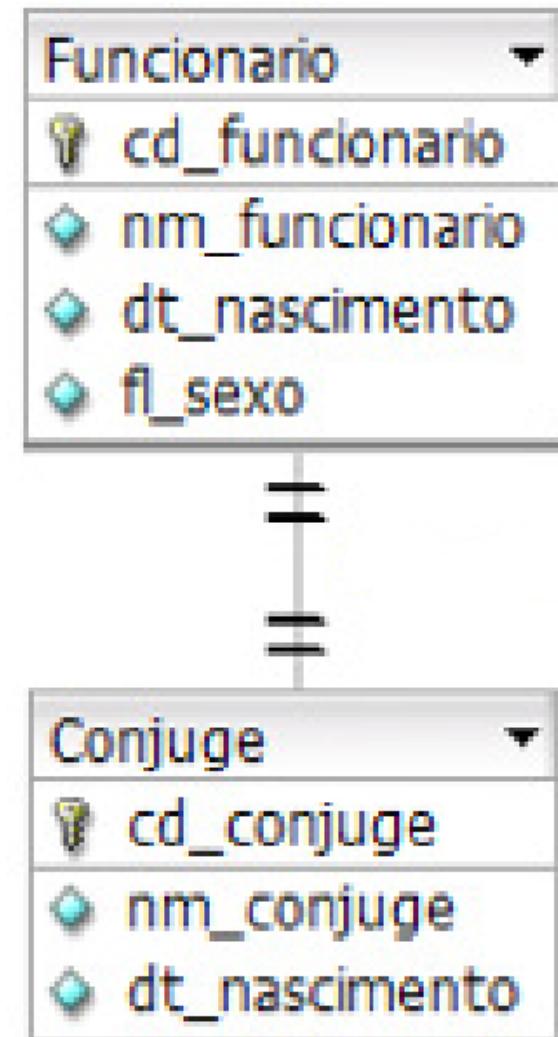
### Summary of Crow's Foot Notation



# Modelagem

## Relacionamento: Um para um

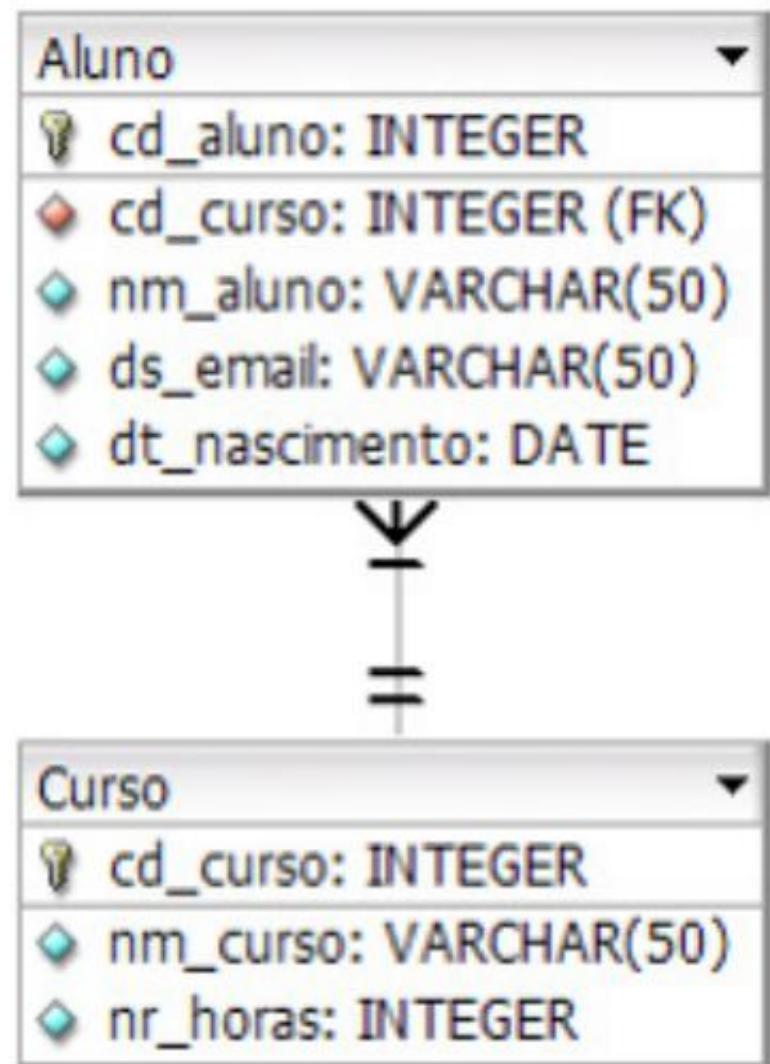
- Ambas tabelas podem ter somente uma instância em ambos os lados
- A maioria das relações “Um para um” são definidas por **regras de negócios**



# Modelagem

## Relacionamento: Um para muitos

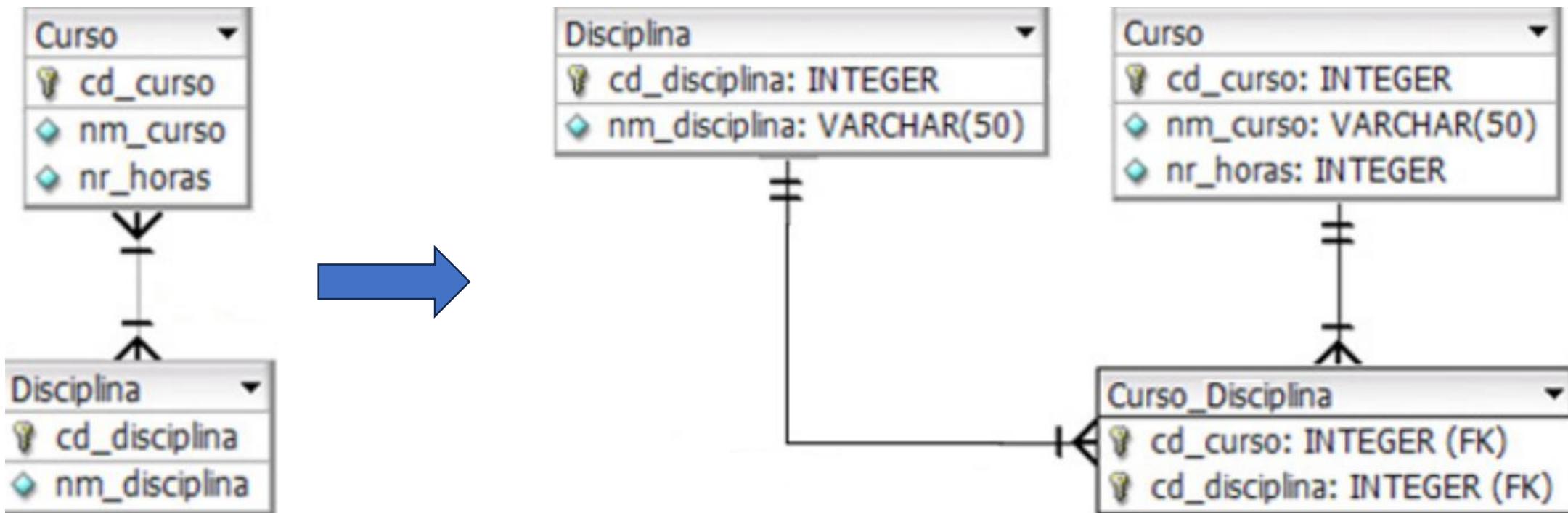
- A chave primária contém somente uma instância que se relaciona a uma ou muitas instâncias da outra tabela
- A chave primária da tabela do "lado 1" se torna uma chave estrangeira na tabela do "lado muitos"



# Modelagem

## Relacionamento: Muitos para muitos

- Cada instância em ambas tabelas podem se relacionar a qualquer número de instâncias em outra tabela
- Esses relacionamentos requerem uma terceira tabela, chamada de tabela de associação



# Modelagem

## Normalização de Dados

- É o processo de organização de um banco de dados para reduzir a redundância e melhorar a integridade dos dados, estruturando os dados em tabelas e definindo relacionamentos entre elas
- O objetivo principal é garantir que o design do banco de dados não tenha dados duplicados entre tabelas e suporte operações eficientes de consulta e atualização, minimizando anomalias de dados
- Segue um conjunto de regras chamadas **formas normais (FN)**

# Modelagem

## 1ª Forma Normal (1FN)

- Também chamada de regra da atomicidade
- Cada célula da tabela deve conter apenas um único valor
- Cada coluna deve ter um nome único
- Ajuda a eliminar dados duplicados, simplificar consultas e reduzir erros de inserção

Students

| FirstName | LastName | Knowledge      |
|-----------|----------|----------------|
| Thomas    | Mueller  | Java, C++, PHP |
| Ursula    | Meier    | PHP, Java      |
| Igor      | Mueller  | C++, Java      |

Startsituation

Result after Normalisation

Students



| FirstName | LastName | Knowledge |
|-----------|----------|-----------|
| Thomas    | Mueller  | C++       |
| Thomas    | Mueller  | PHP       |
| Thomas    | Mueller  | Java      |
| Ursula    | Meier    | Java      |
| Ursula    | Meier    | PHP       |
| Igor      | Mueller  | Java      |
| Igor      | Mueller  | C++       |

# Modelagem

## 2ª Forma Normal (2FN)

- Uma tabela está na 2FN se
  - Ela já está na 1FN
  - Todo atributo não chave é totalmente dependente funcionalmente da chave primária

| Electric toothbrush models |             |                      |
|----------------------------|-------------|----------------------|
| Manufacturer               | Model       | Manufacturer country |
| Forte                      | X-Prime     | Italy                |
| Forte                      | Ultraclean  | Italy                |
| Dent-o-Fresh               | EZbrush     | USA                  |
| Brushmaster                | SuperBrush  | USA                  |
| Kobayashi                  | ST-60       | Japan                |
| Hoch                       | Toothmaster | Germany              |
| Hoch                       | X-Prime     | Germany              |



| Electric toothbrush models |             |
|----------------------------|-------------|
| Manufacturer               | Model       |
| Forte                      | X-Prime     |
| Forte                      | Ultraclean  |
| Dent-o-Fresh               | EZbrush     |
| Brushmaster                | SuperBrush  |
| Kobayashi                  | ST-60       |
| Hoch                       | Toothmaster |
| Hoch                       | X-Prime     |

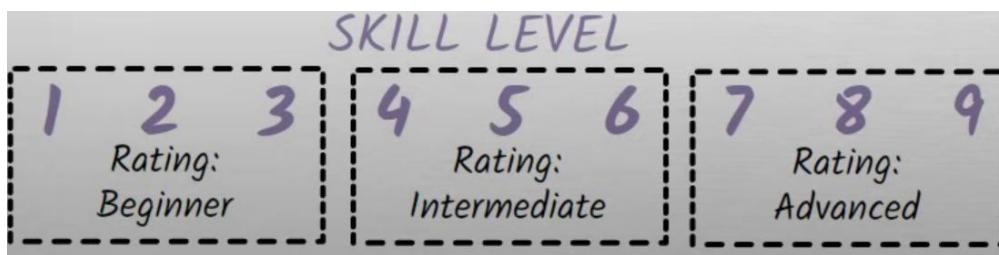
| Electric toothbrush manufacturers |                      |
|-----------------------------------|----------------------|
| Manufacturer                      | Manufacturer country |
| Forte                             | Italy                |
| Dent-o-Fresh                      | USA                  |
| Brushmaster                       | USA                  |
| Kobayashi                         | Japan                |
| Hoch                              | Germany              |

# Modelagem

## 3ª Forma Normal (3FN)

- Uma tabela está na 3FN se
  - Ela já está na 2FN
  - Todos os atributos não-chave dependem diretamente da chave primária, não de outro atributo não-chave

| Player    |               |                    |
|-----------|---------------|--------------------|
| Player_ID | Player_Rating | Player_Skill_Level |
| jdog21    | Intermediate  | 4                  |
| gilal9    | Beginner      | 3                  |
| trev73    | Advanced      | 8                  |
| tina42    | Beginner      | 1                  |



| Player    |                    | Player_Skill_Levels |               |
|-----------|--------------------|---------------------|---------------|
| Player_ID | Player_Skill_Level | Player_Skill_Level  | Player_Rating |
| jdog21    | 4                  | 1                   | Beginner      |
| gilal9    | 4                  | 2                   | Beginner      |
| trev73    | 8                  | 3                   | Beginner      |
| tina42    | 1                  | 4                   | Intermediate  |
|           |                    | 5                   | Intermediate  |
|           |                    | 6                   | Intermediate  |
|           |                    | 7                   | Advanced      |
|           |                    | 8                   | Advanced      |
|           |                    | 9                   | Advanced      |