



Escola Superior de Tecnologia e Gestão
Instituto Politécnico da Guarda



Escola Superior de Tecnologia e Gestão
Instituto Politécnico da Guarda

MODELO RELACIONAL (ORACLE E MYSQL)

José Carlos Fonseca, 2015



O MODELO ER

- ❑ **Modelo Entidade-Relacionamento (Modelo ER)**
permite descrever os dados de um problema real em termos de objetos, dos seus atributos e dos seus relacionamentos
- ❑ Permite passarmos de uma descrição informal do que se pretende da BD para uma descrição mais detalhada e precisa, que pode ser implementada num SGBD



O MODELO ER

EMPREGADOS				
empno	ename	job	hiredate	deptno
7369	Smith	Clerk	1980-12-17	20
7499	Allen	Salesman	1981-02-20	30
7521	Ward	Salesman	1981-02-21	30
7566	Jones	Manager	NULL	20
7782	Clark	Manager	1981-06-09	10

A compreensão do Modelo ER e o relacionamento entre tabelas é feito intuitivamente, mesmo para alguém sem experiência em BD

DEPARTAMENTOS		
deptno	dname	loc
10	Accounting	New York
20	Research	Dallas
30	Sales	Chicago
40	Operations	Boston



O MODELO ER

- ❑ Há diversas notações para representar o **Modelo Entidade-Relacionamento** (Modelo ER)
- ❑ A Oracle adopta o “CASE*Method” (posteriormente chamado “Custom Development Method”). Este modelo segue a notação apresentada em:
 - ❑ Barker, Richard. 1990. *CASE*Method: Entity Relationship Modeling*. Wokingham, England: Addison-Wesley
- ❑ O MySQL com o MySQLWorkbench adopta outros modelos
 - ❑ Para os objetos: Workbench (default), Workbench (simplified), Classic, IDEF1X
 - ❑ Para os relacionamentos: Crow’s Foot (IE), Classic, Connect to columns, UML, IDEF1X



O MODELO ER

- ❑ O Modelo ER é constituído por Entidades, Atributos e Relacionamentos
- ❑ A elaboração de um Modelo ER deve seguir uma abordagem Top-Down:
 1. Entidades, eventualmente com alguns atributos relevantes
 2. Relacionamentos entre entidades
 3. Atributos
 4. Refinar o modelo corrigindo entidades, relacionamentos e atributos



ENTIDADE

- ❑ **Entidade** é algo relevante, que pode ser um objeto real ou imaginário, que é distinto de outros objetos e sobre o qual é necessário guardar informação
- ❑ É descrita por uma série de pelo menos dois **Atributos**, que são atômicos em termos de BD (pois se estes Atributos tiverem eles próprios Atributos então poderão ser Entidades)
- ❑ Se há uma lista de coisas semelhantes, então essas coisas são provavelmente uma Entidade



ENTIDADE

- ❑ No diagrama as Entidades apresentam-se como rectângulos de cantos arredondados
- ❑ Exemplos de entidades:
 - ❑ Alunos
 - ❑ Consultas
 - ❑ Departamentos
 - ❑ Filmes
 - ❑ Artigos
 - ❑ Clientes
 - ❑ ...



ATRIBUTO

- ❑ **Atributo** é um aspecto ou propriedade que descreve uma Entidade. Os Atributos irão representar as colunas das tabelas
- ❑ Os Atributos são atômicos em termos de BD, pois se tiverem atributos então poderão ser Entidades
- ❑ Os Atributos são representados pelo seu Nome e Tipo. O Nome é um substantivo
- ❑ Cada **Atributo** tem um Domínio dos valores possíveis. O Domínio pode não ser representado no modelo lógico (ex. não é incluído na notação usada pela Oracle) sendo normalmente representado no modelo físico (ou relacional)



ATRIBUTO

- ❑ No diagrama os Atributos apresentam-se como nomes dentro das Entidades
- ❑ Exemplos de atributos:
 - ❑ Nome
 - ❑ Morada
 - ❑ Descrição
 - ❑ Data de nascimento
 - ❑ Preço
 - ❑ Altura
 - ❑ ...



IDENTIFICADOR ÚNICO E CHAVE PRIMÁRIA

- ❑ **Identificador Único** define univocamente cada tuplo da Entidade
- ❑ Pode ser constituído por um único atributo ou um conjunto de atributos (que dará origem a uma Chave Primária Composta)
- ❑ Havendo mais do que um (ou um grupo) atributo que define univocamente cada tuplo da Entidade irão dar origem a **Chaves Candidatas**
 - ❑ Uma das Chaves Candidatas será designada **Chave Primária**
 - ❑ Ela é a resultante da implementação no Modelo Físico (ou Relacional) do Identificador Único
- ❑ No modelo lógico não há Chaves Primárias, mas sim Identificadores Únicos



IDENTIFICADOR ÚNICO E CHAVE PRIMÁRIA

- ❑ Características do Identificador Único:
 - ❑ **Unívoco** – cada valor do Identificado Único é único na entidade já que define univocamente cada tuplo da entidade
 - ❑ **Não Nulo** – o valor do Identificado Único não pode ser nulo, nem nenhum dos seus componentes (no caso de dar origem a uma chave composta) o pode ser
- ❑ Às vezes é necessário criar um novo atributo especificamente para ser Identificador Único, ex. quando os atributos existentes não servem, são muitas vezes alterados, têm uma grande dimensão, questões de privacidade, facilitar os joins, etc. A este novo atributo, quando implementado, é dado o nome Surrogate Key



IDENTIFICADOR ÚNICO E CHAVE PRIMÁRIA

- ❑ Os seus atributos do Identificador Único são marcados com um # (Oracle) ou uma 🔑 (MySQL)
- ❑ Exemplos:
 - ❑ Código de Aluno
 - ❑ Número de Fatura
 - ❑ Matrícula do Automóvel
 - ❑ Código de Barras
 - ❑ Número de Encomenda
 - ❑ ...
- ❑ Números pessoais como N° Contribuinte ou N° BI não devem ser usados para este fim devido a questões de privacidade (poderão aparecer como Chaves Estrangeiras noutras Tablas)



CHAVE ESTRANGEIRA

- ❑ **Chave Estrangeira** de uma Tabela é o atributo (ou conjunto de atributos) que aparece como chave primária de uma outra Tabela (ou até da mesma Tabela)
- ❑ É a representação no Modelo Físico (ou Relacional) do Relacionamento do Modelo Lógico
- ❑ Ela não é tipicamente mostrada no diagrama lógico, já que simplesmente resulta da implementação do relacionamento e está implicitamente representada por esse relacionamento
- ❑ Ao contrário da Chave Primária, a Chave Estrangeira normalmente não é unívoca e pode ter valores nulos

TIPOS DE ATRIBUTO

- ❑ Os vários tipos de atributo podem depender do SGBD, mas normalmente são:
 - ❑ Identificador Único, que dá origem à Chave Primária aparece no modelo Físico (ou Relacional)
 - ❑ Atributo Obrigatório (not null)
 - ❑ Atributo Opcional (null)
 - ❑ Atributo Único (unique)
 - ❑ Chave Estrangeira, que é representada normalmente no modelo Físico (ou Relacional)
 - ❑ Atributo Auto Incrementado, que poderá ser representado normalmente no modelo Físico (ou Relacional)
- ❑ Nota: Nem todos estes atributos são representados no modelo Lógico e/ou no modelo Físico (ou Relacional)



RESTRIÇÕES DE INTEGRIDADE

- ❑ Uma base de dados está num estado de integridade se contém apenas dados válidos (estão de acordo com a realidade)
- ❑ Restrições de Integridade:
 - ❑ **Entidade** – Chave Primária e Unique (valores únicos e not null no caso da Chave Primária)
 - ❑ **Referencial** – Chave Estrangeira (só pode tomar valores da Chave Primária)
 - ❑ **Domínio** – Tipo de dados, tamanho, conjunto de valores, limites, obrigatoriedade
 - ❑ **Regras Complexas de Negócio** – Todas as outras restrições, ex. comparações com valores de outros registos: nenhum funcionário pode ganhar mais do que o patrão



RELACIONAMENTO

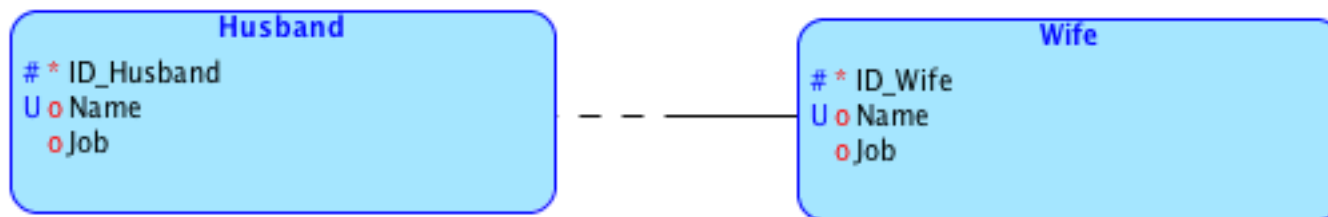
- ❑ Um **Relacionamento** é uma associação entre duas Entidades
- ❑ O Relacionamento é caracterizado por:
 - ❑ **Nome** – e Inglês deve ser uma preposição e não um verbo. Em Português um verbo funciona melhor
 - ❑ **Participação** – Indica se o relacionamento é Obrigatório ou não (Not Null ou Null)
 - ❑ **Cardinalidade** – Indica o grau do relacionamento
 - ❑ Um para Um
 - ❑ Um para Muitos
 - ❑ Muitos para Muitos



MODELO LÓGICO E MODELO FÍSICO

- ❑ Após o desenho do Modelo ER (Modelo Lógico) este tem de se implementar num Modelo Físico
- ❑ Modelo Lógico - Modelo abstracto, independente da tecnologia em que vai ser implementado, desenvolvido por Peter Chen em 1976 que pode ser implementado numa qualquer BD
 - ❑ Entidades
 - ❑ Atributos
 - ❑ Relacionamentos
- ❑ Modelo Físico (ou Relacional) – Instanciação do Modelo Lógico numa BD específica, seguindo as regras próprias dessa BD
 - ❑ Tabelas (nem cada Entidade dá origem a uma e uma só Tabela)
 - ❑ Campos ou Colunas (ter em atenção aos tipos de dados e restrições)
 - ❑ Relacionamentos (chaves estrangeiras)

RELACIONAMENTO DE UM PARA UM

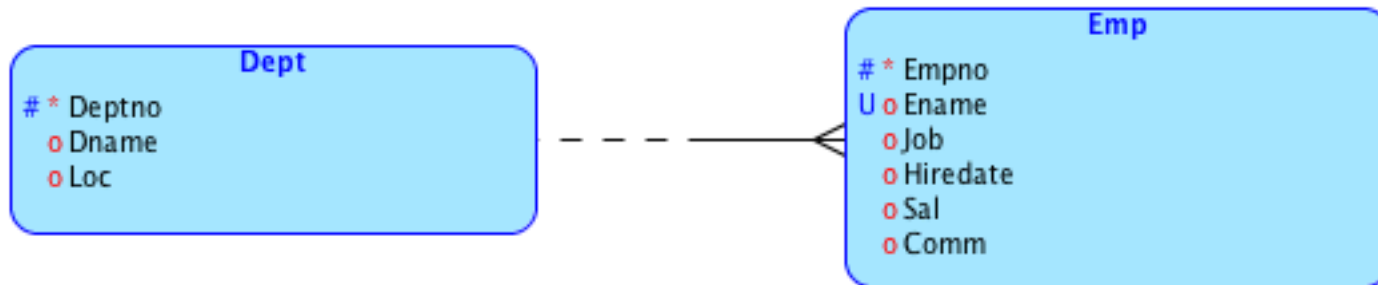


É muito raro acontecer na prática

Pode dar origem a duas tabelas, mas normalmente dá origem a uma única tabela com o conjunto dos campos das outras duas tabelas

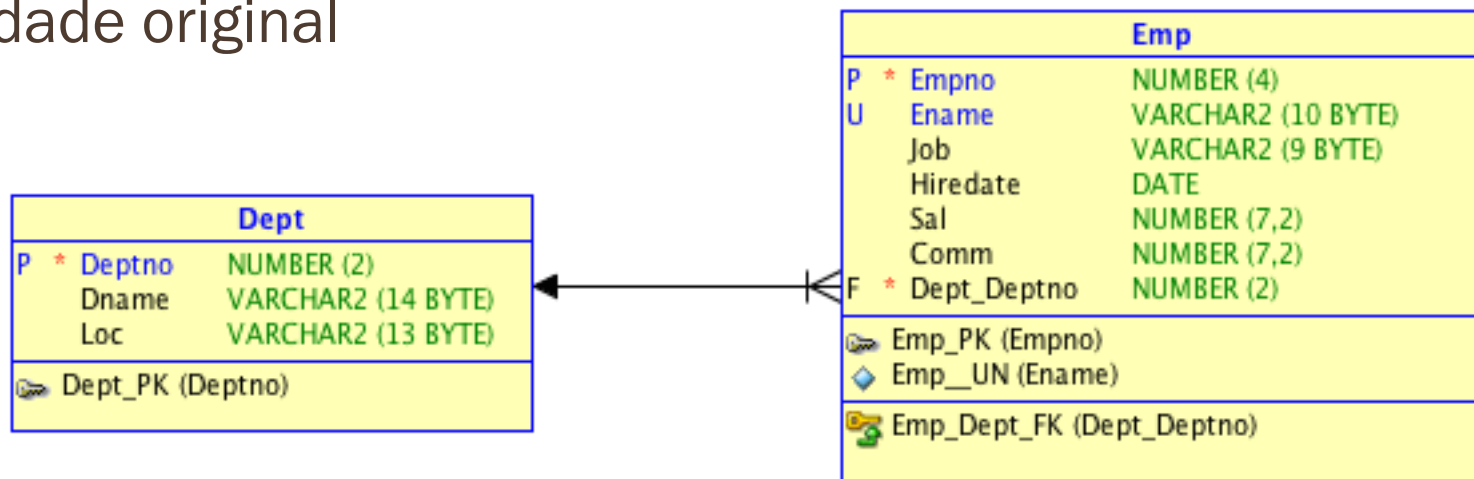
Couple		
P	* ID_Couple	NUMBER (4)
U	Name_Husband	VARCHAR2 (10 BYTE)
	Job_Husband	VARCHAR2 (9 BYTE)
U	Name_Wife	VARCHAR2 (10 BYTE)
	Job_Wife	VARCHAR2 (9 BYTE)
Couple_PK (ID_Couple) Couple__UN (Name_Husband) Couple__UNv1 (Name_Wife)		

RELACIONAMENTO DE UM PARA MUITOS



Este é, de longe, o grau de Relacionamento mais comum. De facto, quando num ER aparecem outros tipos de Relacionamento, estes devem ser verificados para ver se não houve erro de análise

Dá origem a duas tabelas, correspondendo cada uma a cada entidade original

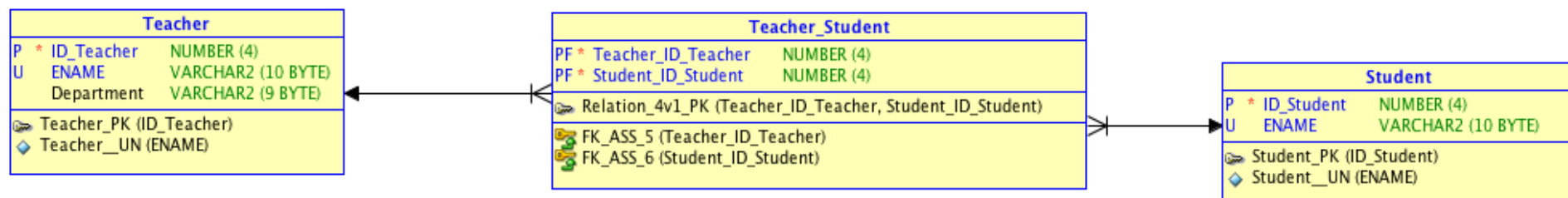


RELACIONAMENTO DE MUITOS PARA MUITOS



É pouco comum acontecer na prática

Cada Tabela fica com o conjunto dos campos das outras duas tabelas. É criada uma outra tabela de ligação contendo uma chave estrangeira para cada uma das tabelas. O conjunto dessas chaves estrangeiras pode formar uma chave primária ou pode haver mais uma coluna que será a chave primária sintética (surrogate key)

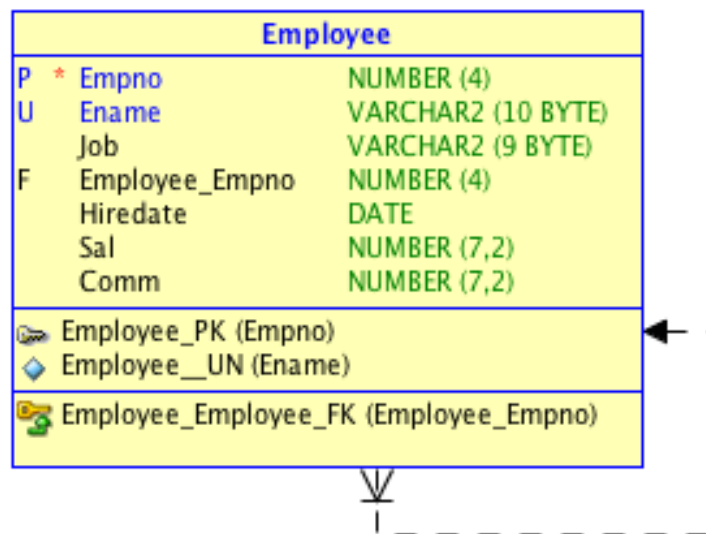
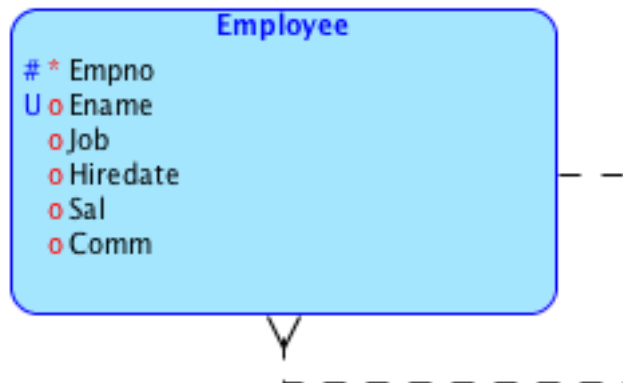




RELACIONAMENTO DE MUITOS PARA MUITOS

- ❑ Tabela intermédia de ligação cuja chave primária é formada pelo conjunto das chaves primárias das outras tabelas
 - ❑ Ex. Faturas, Linhas de Fatura, Produtos NÃO PODENDO haver na mesma fatura linhas de fatura com o mesmo produto
- ❑ Tabela intermédia de ligação cuja chave primária é uma chave sintética (surrogate key), havendo uma chave estrangeira para cada uma das chaves primárias das outras tabelas
 - ❑ Ex. Faturas, Linhas de Fatura, Produtos PODENDO haver na mesma fatura linhas de fatura com o mesmo produto

RELACIONAMENTO RECURSIVO



Pode haver relacionamentos de um para muitos ou de muitos para muitos

Executam-se os mesmos procedimentos anteriormente descritos para os vários graus de Relacionamento, embora neste caso as duas tabelas sejam, na realidade, a mesma tabela



SUBTIPOS DE ENTIDADES

- ❑ Subtipos de Entidades. No caso de haver uma entidade genérica da qual dependem outras entidades.
 - ❑ Ex. Entidade Pessoa da qual dependem as Entidades Médico, Doente, Enfermeiro por terem atributos comuns
- ❑ Soluções para implementação do Modelo Físico (ou Relacional):
 - ❑ **Roll-up** – Todas as Entidades dão origem a uma única Tabela
 - ❑ **Roll-down** – Cada Tabela fica com todos os atributos que necessita da Entidade correspondente juntamente com a Entidade genérica. A Entidade genérica poderá ser eliminada, se não houver mais necessidade dela
 - ❑ **Múltiplas tabelas** – Cada Entidade fica com os seus atributos, ficando a tabela geral relacionada com as outras

ENTIDADES FRACAS

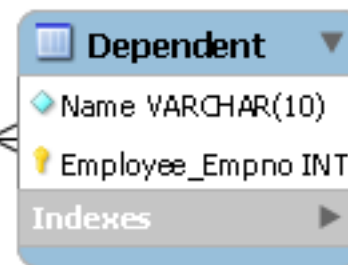
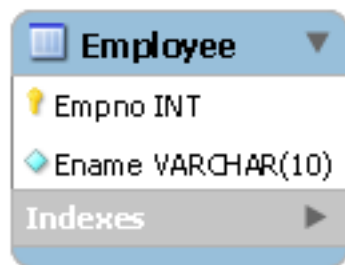
- As entidades fracas não têm, de forma natural, chave primária formada por atributos próprios
- A chave primária depende sempre de outra entidade, designada por entidade identificadora

Entidade Identificadora

Employee		
P *	Empno	NUMBER (4)
*	Ename	VARCHAR2 (10)
Employee_PK (Empno)		

Dependent		
	Nome	VARCHAR2 (10)
PF *	Employee_Empno	NUMBER (4)
Dependent_PK (Employee_Empno)		
Dependent_Employee_FK (Employee_Empno)		

Entidade Fraca





Escola Superior de Tecnologia e Gestão
Instituto Politécnico da Guarda



Escola Superior de Tecnologia e Gestão
Instituto Politécnico da Guarda

EXEMPLOS COM ORACLE (SQL DEVELOPER)

José Carlos Fonseca, 2015

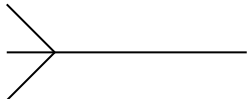


TIPOS DE ATRIBUTO (ORACLE)

- ❑ Os vários tipos de atributo (notação Oracle) são:
 - ❑ # Identificador Único, Unique Identifier (UID)
 - ❑ * Atributo Obrigatório, Mandatory Attribute (Not Null)
 - ❑ o Atributo Opcional, Optional Attribute (Null)
 - ❑ U Atributo Único, Unique Attribute












RELACIONAMENTO (ORACLE)

Terminador Relacional	Leitura	Notação
Opcional	pode	-----
Obrigatório	tem de	_____
Cardinalidade = 1	um e um só	_____
Cardinalidade ≥ 1	um ou vários	



RELACIONAMENTO (ORACLE)

-  M:1 (Obrigatório para Opcional)
-  M:1 (Opcional para Opcional)
-  M:1 (Opcional para Obrigatório)
-  M:1 (Obrigatório para Obrigatório)
-  M:M (Opcional para Opcional)
-  M:M (Obrigatório para Opcional)
-  1:1 (Obrigatório para Opcional)
-  1:1 (Opcional para Opcional)
-  1:1 (Obrigatório para Obrigatório)



RELACIONAMENTO (ORACLE)

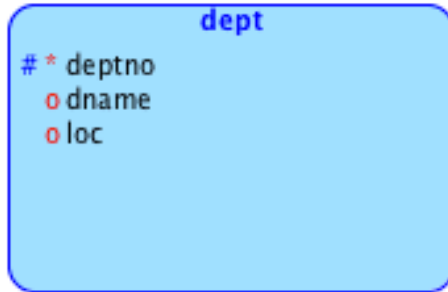
□ O Relacionamento deve ler-se da seguinte maneira:

1. Cada
2. <entidade 1>
3. {tem de | pode} (se a linha é contínua ou tracejada)
4. <relacionamento>
5. {Um ou Vários | Um e só Um} (se tem ou não um pé de galinha)
6. <entidade 2>

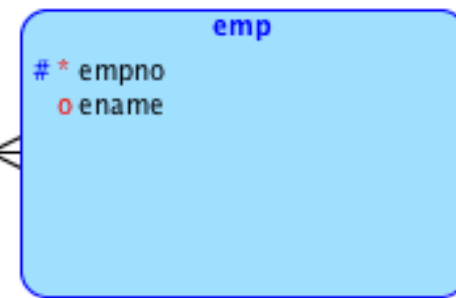


RELACIONAMENTO (ORACLE)

Source/Pai (lado do um)



Target/Filho (lado do muitos)



Relation Properties - FK_DEPTNO

General

Name: FK_DEPTNO

Use surrogate keys: ☐

Source Cardinality

Source: dept

Source key: dept.dept PK

Name on Source:

Source Entity Synonym: dept

Source to Target Cardinality: 1 * (indicated by a line with an arrow pointing to the asterisk)

Source Optional: ☒

Transferable: ☒

Dominant Role: None

Identifying: ☐

Delete Rule: NO ACTION

Target Cardinality

Target: emp

Target key:

Name on Target:

Target Entity Synonym: emp

Target to Source Cardinality: 1 (indicated by a line with an arrow pointing to the '1')

Target Optional: ☐

Transferable: ☒

Buttons: OK, Apply, Cancel, Help

“Cada Departamento
pode ter um ou vários
Empregados”

e

“Cada Empregado tem
de pertencer a um e um
só Departamento”

MODELO RELACIONAL (ORACLE)

Source (lado do um)

Target (lado do muitos)

dept	
P *	deptno NUMBER (2)
	dname VARCHAR2 (14)
	loc VARCHAR2 (13)
dept_PK (deptno)	



emp	
P *	empno NUMBER (4)
	ename VARCHAR2 (10)
F *	dept_deptno NUMBER (2)
emp_PK (empno)	
emp_dept_FK (dept_deptno)	

“Cada Departamento pode ter um ou vários Empregados”
e

“Cada Empregado tem de pertencer a um e um só Departamento”



Escola Superior de Tecnologia e Gestão
Instituto Politécnico da Guarda








Escola Superior de Tecnologia e Gestão
Instituto Politécnico da Guarda

EXEMPLOS COM MYSQL (MYSQL WORKBENCH)

José Carlos Fonseca, 2015

TIPOS DE ATRIBUTO (ORACLE E MYSQL)

- ❑ Os vários tipos de atributo (notação MySQL) são:
 - ❑  Chave Primária
 - ❑  Atributo Obrigatório (Not Null)
 - ❑  Atributo Obrigatório Chave Estrangeira (Not Null)
 - ❑  Atributo Opcional (Null)
 - ❑  Atributo Opcional Chave Estrangeira (Null)



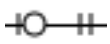



RELACIONAMENTO (MYSQL)

Terminador Relacional	Leitura	Notação
Opcional	pode	○
Obrigatório	tem de	
Cardinalidade = 1	um e um só	+ — +
Cardinalidade ≥ 1	um ou vários	> — +
Identifying		———
Non Identifying		-----



RELACIONAMENTO (MYSQL)

-  M:1 (Obrigatório para Opcional)
-  M:1 (Opcional para Opcional)
-  1:1 (Opcional para Obrigatório)
-  1:1 (Obrigatório para Obrigatório)

RELACIONAMENTO (MYSQL)

□ O Relacionamento deve ler-se da seguinte maneira:

1. Cada
2. <entidade 1>
3. {tem de | pode} (se tem um | ou um ○)
4. <relacionamento>
5. {Um ou Vários | Um e só Um} (se tem ou não um pé de galinha)
6. <entidade 2>

MODELO RELACIONAL (MYSQL)

Source (lado do um)

Target (lado do muitos)



“Cada Departamento pode ter um ou vários Empregados”
e

“Cada Empregado tem de pertencer a um e um só Departamento”

MODELO RELACIONAL (MYSQL)

Source (lado do um)

Target (lado do muitos)



“Cada Departamento pode ter um ou vários Empregados”
e

“Cada Empregado pode pertencer a um e um só Departamento”