

MODELO RELACIONAL

- A próxima etapa do projeto de banco de dados envolve o chamado **modelo lógico**.

MODELO RELACIONAL

- Atualmente, grande parte dos sistemas de banco de dados utiliza o **modelo relacional**.
- Um banco de dados relacional é composto por tabelas (também denominadas **relações**).

MODELO RELACIONAL

Observe a seguir alguns conceitos importantes para pleno entendimento do modelo relacional:

TABELA

Estrutura bi-dimensional composta por **linhas** (tuplas) e **campos** (ou atributos).

The diagram illustrates a table structure with the following components labeled:

- nome da tabela** (table name): Indicated by an orange line pointing to the header cell 'Departamento'.
- campo ou atributo** (field or attribute): Indicated by a green line pointing to the header cell 'CodDept'.
- linha ou tupla** (line or tuple): Indicated by a red line pointing to the first data row containing 'D001' and 'Financeiro'.
- coluna** (column): Indicated by a blue line pointing to the 'Nome' column.

Departamento	
CodDept	Nome
D001	Financeiro
D002	Engenharia
D003	Comercial

CHAVE PRIMÁRIA (PK - PRIMARY KEY)

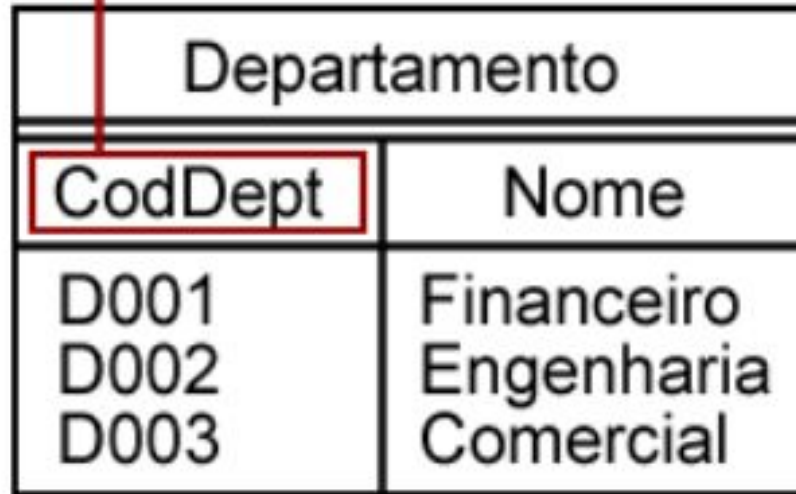
- Atributo através do qual seja possível **identificar determinado registro (ocorrência)**.
- Uma chave primária não pode ser repetida, ou seja, o conjunto de valores que constituem a chave primária **deve ser único** dentro de uma tabela.

CHAVE PRIMÁRIA (PK - PRIMARY KEY)

- **Chave Primária Simples:** apenas um atributo (campo) compõe a chave primária.
- **Chave Primária Composta:** mais de um atributo compõe a chave primária.

CHAVE PRIMÁRIA (PK - PRIMARY KEY)

chave primária (PK - Primary Key)



A diagram illustrating a primary key. A red line points from the text 'chave primária (PK - Primary Key)' to the 'CodDept' column header in a table. The table has two columns: 'CodDept' and 'Nome'. The 'CodDept' column contains three rows of data: 'D001', 'D002', and 'D003'. The 'Nome' column contains three rows of data: 'Financeiro', 'Engenharia', and 'Comercial'.

Departamento	
CodDept	Nome
D001	Financeiro
D002	Engenharia
D003	Comercial

CHAVE ÚNICA (UNIQUE)

- Utilizada quando determinado campo não deve ser repetido e não é chave primária.
 - Aumenta consistência do banco de dados.

CHAVE ÚNICA (UNIQUE)

Exemplo: Cadastro de funcionários.

- Cada funcionário recebe um **código único**, que é a chave primária.
- Para maior segurança e consistência podemos optar que o campo **CPF** seja único, evitando que o mesmo funcionário seja cadastrado duas vezes.

CHAVE ÚNICA (UNIQUE)

chave única (unique)

Funcionario		
CodFunc	Nome	CPF
1001	Antonio	000111222-33
1002	Beatriz	111222333-44
1003	Cláudio	222333444-55

CHAVE ESTRANGEIRA (FK - FOREIGN KEY)

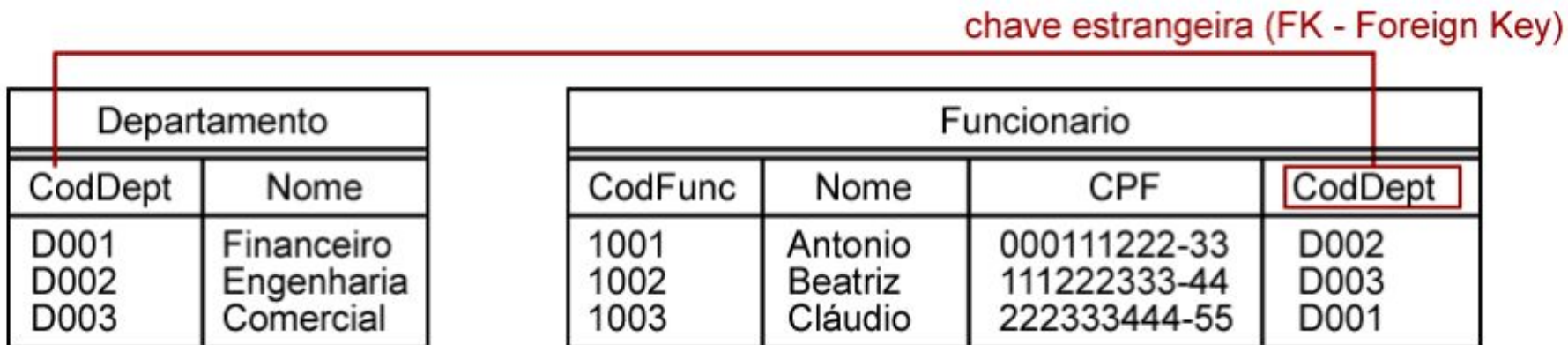
- Utilizada quando queremos que o valor de um atributo seja validado a partir do valor de atributo de outra tabela.
- Criamos assim uma **relação de dependência** (um relacionamento) entre as tabelas.

CHAVE ESTRANGEIRA (FK - FOREIGN KEY)

Exemplo:

- Antes de efetuar a alocação de um funcionário em um departamento, é necessário que o departamento em questão conste na tabela de departamentos.

CHAVE ESTRANGEIRA (FK - FOREIGN KEY)



RELACIONAMENTOS

- Associação estabelecida entre **campos comuns** de **duas tabelas**.
- Dessa forma permitimos o estabelecimento de correspondência entre registros de diferentes tabelas.

RELACIONAMENTOS

- Os relacionamentos apresentam a seguinte **classificação** quanto à sua cardinalidade:

RELACIONAMENTO UM-PARA-UM (1:1)

Cada ocorrência de uma tabela relaciona-se com uma e somente uma ocorrência da outra tabela.

RELACIONAMENTO UM-PARA-MUITOS (1:N)

Uma ocorrência da tabela pai relaciona-se com muitas ocorrências da tabela filho, mas cada ocorrência da tabela filho somente pode estar relacionada com uma ocorrência da tabela pai.

RELACIONAMENTO MUITOS-PARA-MUITOS (N:N)

Apresenta em ambos os sentidos um ou mais relacionamentos de um-para-muitos.

RELACIONAMENTO MUITOS-PARA-MUITOS (N:N)

No modelo relacional **não é possível** efetuar este tipo de relacionamento de forma direta.

Neste caso, **deve-se construir uma terceira tabela** (tabela de associação ou tabela de detalhes).

RELACIONAMENTO MUITOS-PARA-MUITOS (N:N)

Essa tabela deve possuir **chave primária composta de dois campos** (as chaves estrangeiras provenientes das duas tabelas originais).

RELACIONAMENTO MUITOS-PARA-MUITOS (N:N)

Concluindo, um relacionamento de **muitos-para-muitos** deve ser dividido em dois relacionamentos de um-para-muitos com uma terceira tabela.

NOTAÇÃO RESUMIDA PARA MODELOS LÓGICOS RELACIONAIS

Notação **compacta**, útil para discussões sobre a estrutura geral do banco de dados, utilizada quando não se deseja entrar no nível maior de detalhamento.

NOTAÇÃO RESUMIDA PARA MODELOS LÓGICOS RELACIONAIS

Observe o exemplo a seguir:

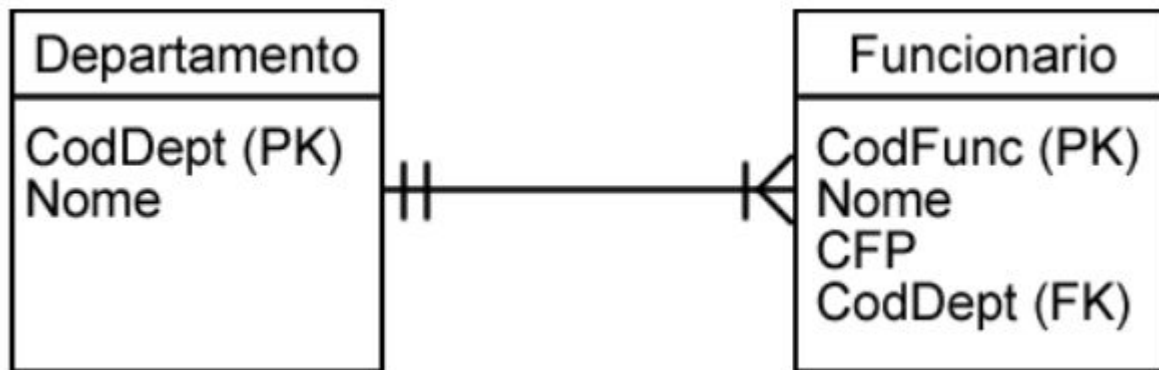
Departamento (CodDept, Nome)

Funcionario (CodFunc, Nome, CPF, CodDept)

CodDept referencia Departamento

NOTAÇÃO RESUMIDA PARA MODELOS LÓGICOS RELACIONAIS

A notação resumida acima representa o seguinte
relacionamento entre as tabelas Departamento e Funcionário:



NOTAÇÃO RESUMIDA PARA MODELOS LÓGICOS RELACIONAIS

Observe que através da notação resumida **não é possível determinar se o relacionamento é do tipo 1:1 ou 1:N** (como no caso representado na figura).

INTEGRIDADE DE DADOS

- Impor a integridade de dados garante a **qualidade** dos dados em um banco de dados.
- Os dados devem refletir corretamente a realidade representada pelo banco e também devem ser consistentes entre si.

INTEGRIDADE DE DOMÍNIO

- Zela pelos valores ideais e necessários para um atributo.
- Para isso definimos algumas regras de validação por meio de expressões compostas de valores constantes.

INTEGRIDADE DE DOMÍNIO

Exemplos:

- Não permitir um estoque negativo.
- Impedir uma data de nascimento superior à data atual.
- Não permitir que o valor de um produto seja negativo.

INTEGRIDADE DE ENTIDADE

- Tem o objetivo de validar os valores permitidos a partir de valores já inseridos na própria entidade.
- Após uma “auto-consulta” a entidade vai permitir ou não a gravação do novo registro.

INTEGRIDADE DE ENTIDADE

Exemplos:

- Não permitir duas pessoas com o mesmo CPF.
- Impedir a locação uma fita que já está locada.

INTEGRIDADE REFERENCIAL

- Zela pela consistência dos registros de uma entidade a partir de valores provenientes de outras entidades, isto é, determinado registro vai “depende” diretamente de um registro de outra tabela.

INTEGRIDADE REFERENCIAL

Exemplos:

- Um registro em uma tabela pai pode ter um ou mais registros em uma tabela filho.
- Um registro em uma tabela filho sempre tem um registro coincidente em uma tabela pai.

INTEGRIDADE REFERENCIAL

Exemplos:

- Para a inclusão de um registro em uma determinada tabela filho, é necessário que exista um registro pai coincidente.
- Um registro pai só poderá ser excluído se não possuir nenhum registro filho.

RESTRIÇÕES (CONSTRAINTS)

- Observe a seguir as principais **constraints** ou **restrições** utilizadas nos bancos de dados relacionais, principalmente durante o processo de criação das tabelas, para implementar os tipos de integridade anteriormente descritos:

RESTRIÇÕES (CONSTRAINTS)

TIPO	ARMAZENA
PRIMARY KEY	Identifica a chave primária da tabela
FOREIGN KEY	Identifica a chave estrangeira
UNIQUE	Indica que os valores na coluna não podem ser repetidos
CHECK	Especifica os valores que uma coluna pode assumir
NOT NULL	Indica que o campo não pode receber valores nulos

NOMENCLATURA DE TABELAS E DE CAMPOS

- Os sistemas gerenciadores de bancos de dados geralmente impõem certas restrições quanto aos caracteres válidos para denominar tabelas, campos (colunas), bem como outros objetos do banco de dados.

NOMENCLATURA DE TABELAS E DE CAMPOS

Observe a seguir o que deve ser evitado:

- Não utilizar caracteres especiais (exceto o underscore “_”).
- Começar com uma letra e não com um número.
- Evitar acentuação e “ç”.
- Não utilizar espaços.

TIPOS DE DADOS

- Durante a criação das tabelas do banco de dados é necessário informar que tipo de dados cada coluna deverá armazenar.
- As tabelas a seguir apresentam alguns tipos de dados compatíveis com o Oracle:

TIPOS DE DADOS

TIPO	ARMAZENA
CHAR	Cadeia de caracteres de tamanho fixo
VARCHAR2	Cadeia de caracteres de tamanho variável
NUMBER	Valores numéricos (inteiros e ponto flutuante)
DATE	Data e hora (inclusive minutos e segundos)