



3. MODELO RELACIONAL

3.1. INTRODUÇÃO

Com o passar dos anos, os SGBD Relacionais tornaram-se o meio mais largamente utilizado para gerenciar os dados, isto porque oferecem fácil acesso aos dados, flexibilidade na modelagem de dados, linguagem de manipulação de dados (SQL) e são de fácil entendimento.

No Modelo Relacional o usuário tem uma visão dos dados na forma de uma coleção de tabelas. O conceito de tabela, embora seja simples e intuitivo, apresenta uma forte correspondência com o conceito matemático de uma relação. Daí a denominação de Modelo Relacional.

Um Modelo de Dados é uma combinação de conjunto de tipos de estruturas de dados (relações), conjunto de operadores (linguagem de banco de dados) e regras de integridade (definidas no esquema do banco de dados e aplicadas às instâncias).

TERMINOLOGIAS

RELAÇÃO	TABELA	ARQUIVO
TUPLA	LINHA	REGISTRO
ATRIBUTOS	COLUNA	CAMPO

Exemplo de relações (tabelas)

CODIGO_MUNICIPIO	NOME_MUNICIPIO	POPULAÇÃO
10	ASSIS	120.000
20	RONDONÓPOLIS	300.000
30	CURITIBA	700.000

- Relação é uma tabela bidimensional;
- Cada linha na tabela contém itens de dados que pertencem a alguma coisa ou a uma parte desta;
- Cada coluna contém dados relativos a um atributo;
- As linhas são também chamadas de tuplas e as colunas de atributos;
- Todos os valores em uma coluna devem ser do mesmo atributo;
- A ordem das colunas e das linhas é irrelevante;
- Cada linha é distinta;
- Os termos relação, tupla e atributo vieram da álgebra relacional, que é a fonte teórica deste modelo.



3.2. MODELAGEM DE DADOS

O Modelo Entidade-Relacionamento (E-R) é um dos mais utilizados para o projeto conceitual de um banco de dados. Esse modelo baseia-se na percepção de um universo constituído por um grupo básico de objetos chamados de entidade e por relacionamentos entre estes objetos.

Esse modelo foi proposto por Peter Chen em 1976 para facilitar o projeto de bancos de dados.

O Modelo E-R representa graficamente o esquema conceitual do banco de dados por meio de Diagramas de Entidade-Relacionamento (Diagramas E-R).

O Diagrama E-R é um dos diagramas mais comumente utilizados para modelar Bancos de Dados Relacionais. A estrutura lógica do banco de dados pode ser expressa graficamente pelo Diagrama E-R. A relativa simplicidade e a clareza gráfica desta técnica de diagramação, explica, em grande parte, a ampla disseminação do uso do Modelo E-R.

Em um Diagrama E-R, os dados são descritos por intermédio de conjuntos de entidades (retângulos); de conjuntos de relacionamentos (losangos); de conjuntos de atributos que compõem as entidades (elipses); e de cardinalidades as quais expressam o número de entidades a que outra entidade pode estar associada, por meio de um conjunto de relacionamentos.

Após a elaboração dos Diagramas E-R, pode-se iniciar a elaboração do modelo físico de dados. O modelo físico de dados deve refletir o que está graficamente representando pelos diagramas.

Segundo DATE (2003, p.358), "Diagramas E-R constituem uma técnica para representar a estrutura lógica de um banco de dados de modo pictórico, e fornecem um meio simples e fácil de entender e para comunicar os aspectos principais do projeto de qualquer banco de dados".

3.2.1 Conjunto de Entidades (Tabelas)

Segundo HEUSER (2000, p.12), uma entidade representa um conjunto de objetos, perceptíveis ou manipuláveis, da realidade modelada sobre os quais se deseja manter informações no banco de dados.



Já conforme KROENKE (1999, p.39), uma entidade é algo que pode ser identificado no ambiente de trabalho do usuário, alguma coisa que ele quer localizar, como por exemplo, a funcionária Mary Doe, o cliente John Smith, a venda número 1000, etc.

Um conjunto de entidades é representado graficamente por um *retângulo*, como ilustrado na Figura 1.



Figura 1. Exemplos de Representação Gráfica de Conjunto de Entidades

Vale ressaltar que os conjuntos de entidades, fisicamente implementados nos Bancos de Dados Relacionais, são conhecidos por *tabelas*, que por sua vez contém um conjunto único de *colunas* (atributos), que é utilizado para armazenar, por meio de *linhas*, as entidades.

Uma entidade é uma “coisa” ou um “objeto” no mundo real que pode ser identificado de forma unívoca em relação a todos os outros objetos; podendo ser concreta, como uma pessoa ou um livro, ou podendo ser abstrata, como um empréstimo ou um conceito; e que contém um conjunto de propriedades, chamados de atributos (SILBERSCHATZ, 1999, p. 21).

Segundo SILBERSCHATZ (1999, p. 21), um conjunto de entidades é um conjunto que abrange entidades do mesmo tipo que compartilham as mesmas propriedades, os atributos.

Por exemplo, o conjunto de todas as pessoas que são empregados de uma empresa, pode ser definido como o conjunto de entidades EMPREGADO. Portanto, de acordo com o autor, as entidades são formadas por um conjunto de propriedades, conhecidas por atributos, e quando se juntam entidades do mesmo tipo, com os mesmos atributos, forma-se um conjunto de entidades.

3.2.2. Conjunto de Propriedades (Colunas)

As entidades possuem atributos, conhecidos também por propriedades, que descrevem as características da entidade, como por exemplo: NOME, EMAIL, CIDADE, NASCIMENTO; ilustrados na Figura 10, que descrevem as características do conjunto de entidades EMPREGADO (KROENKE, 1999, p. 39).



Os atributos ou propriedades são representados graficamente por *elipses*, como ilustrado na Figura 2.

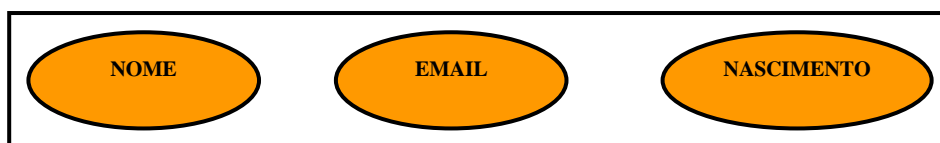


Figura 2. Exemplos de Representação Gráfica de Atributos (Propriedades)

Os atributos, quando implementados nos Bancos de Dados Relacionais, são conhecidos por *colunas*, que são utilizadas para armazenar os valores.

Conforme HEUSER (2003, p.23), um atributo é um dado que é associado a cada ocorrência de uma entidade ou de um relacionamento, conforme ilustrado na Figura 3.

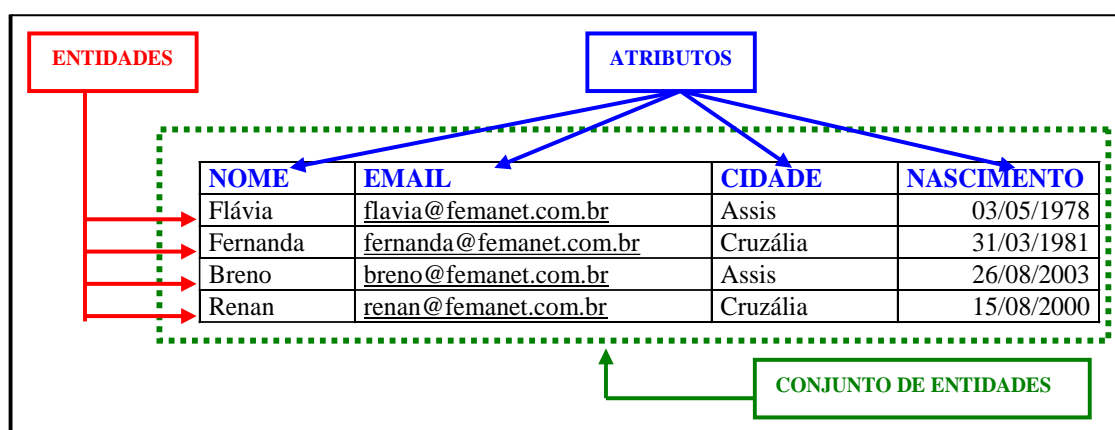


Figura 3. Exemplo de um Conjunto de Entidades

De acordo com a Figura 3, o conjunto de entidades EMPREGADO, é formado pelas seguintes propriedades (atributos): “NOME, EMAIL, CIDADE e NASCIMENTO”, e que pode ser representado da seguinte forma:

EMPREGADO = (NOME, EMAIL, CIDADE, NASCIMENTO)

3.2.3. Conjunto de Relacionamentos

Conforme SILBERSCHATZ (1999, p. 24-25), um relacionamento é uma associação entre uma ou várias entidades, e um conjunto de relacionamento é um conjunto de relacionamentos de mesmo tipo.



Segundo HEUSER (2000, p.13) um relacionamento é um conjunto de associações entre entidades. Por exemplo, um relacionamento EMPREGADO-DEPTO relaciona um empregado ao departamento no qual ele deve estar designado (lotado), como ilustrado na Figura 5.

Os relacionamentos são representados graficamente por *losangos*, como ilustrado na Figura 4.

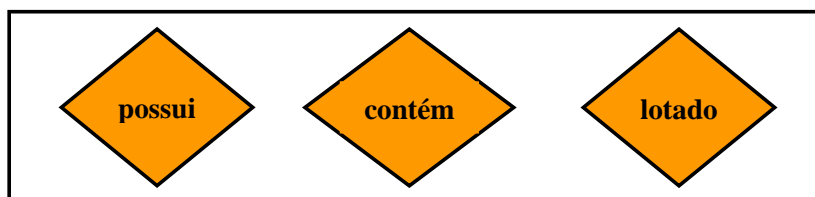


Figura 4. Exemplos de Representação Gráfica de Conjunto de Relacionamentos

É por meio do relacionamento, que é possível relacionar as entidades de um conjunto de entidades, com as entidades de outro conjunto de entidades, formando assim um conjunto de relacionamentos, conforme ilustrado na Figura 5.

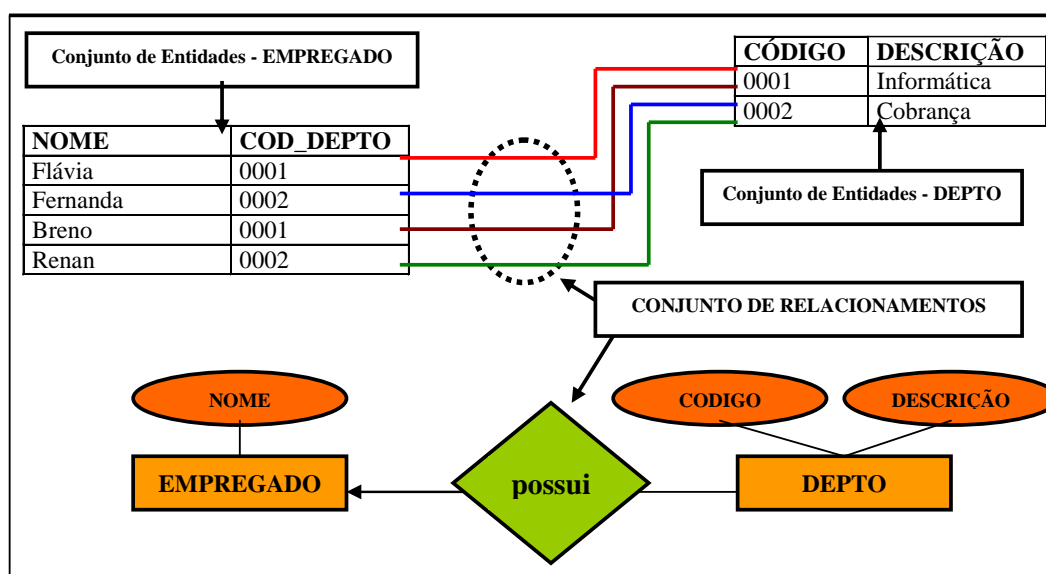


Figura 5. Conjunto de Relacionamentos (EMPREGADO-DEPTO)

3.2.3.1 Cardinalidades (Tipos de Relacionamentos)

A cardinalidade expressa o número de entidades ao qual uma entidade pode estar associada através de um relacionamento.



Um relacionamento binário pode ser:

(1:1) UM-PARA-UM: uma entidade em A está associada com no máximo uma entidade em B, e uma entidade em B está associada com no máximo uma entidade em A.

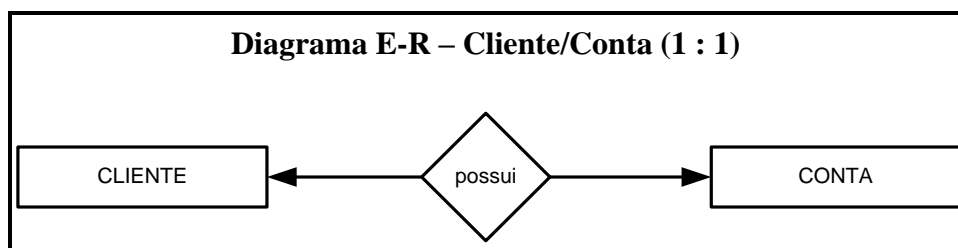


Figura 2. Exemplo 1 de Diagrama E-R (1 : 1)

Um cliente possui uma única conta, e cada conta pertence a um único cliente.

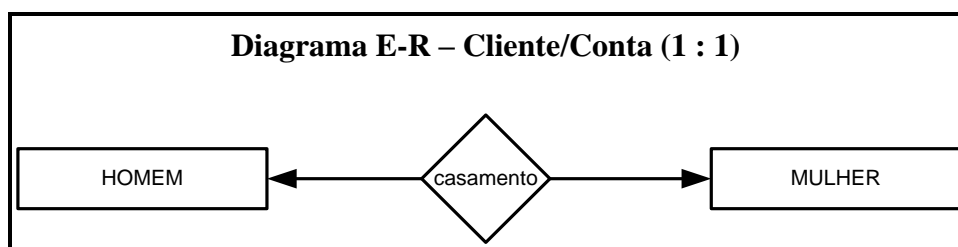


Figura 3. Exemplo 2 de Diagrama E-R (1 : 1)

(1:N) UM-PARA-MUITOS: uma entidade em A está associada com qualquer número de entidades em B, que está associada a no máximo uma entidade em A.

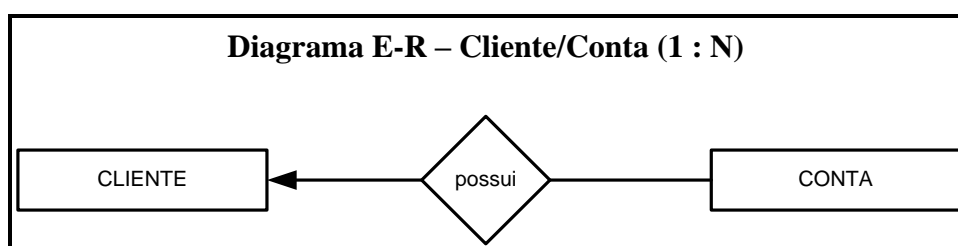


Figura 4. Exemplo 1 de Diagrama E-R (1 : N)



Um cliente possui muitas contas, mas cada conta pertence a um único cliente.

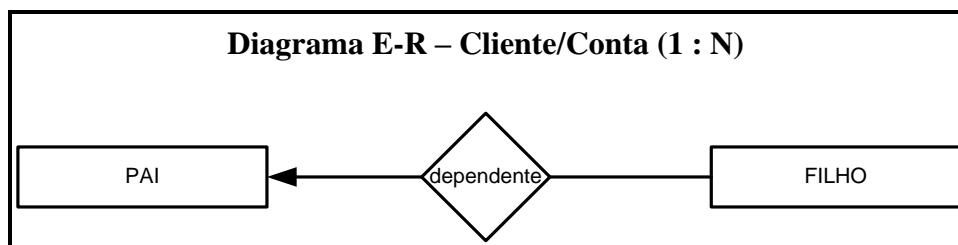


Figura 5. Exemplo 2 de Diagrama E-R (1 : N)

Um pai pode ter muitos filhos, mas cada filho tem apenas um pai.

(N:1) MUITOS-PARA-UM: uma entidade em A está associada com no máximo uma entidade em B, e uma entidade em B está associada com qualquer número de entidades em A. Similar ao relacionamento (1:N). Por exemplo, muitos clientes possuem uma mesma conta (conta conjunta), mas cada conta só pode pertencer a um único cliente.

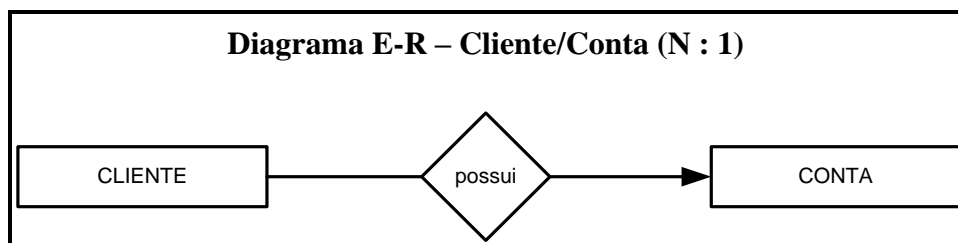


Figura 6. Exemplo de Diagrama E-R (N : 1)

(N:M) MUITOS-PARA-MUITOS: uma entidade em A está associada com qualquer número de entidade em B, e uma entidade em B está associada com qualquer número de entidades em A.

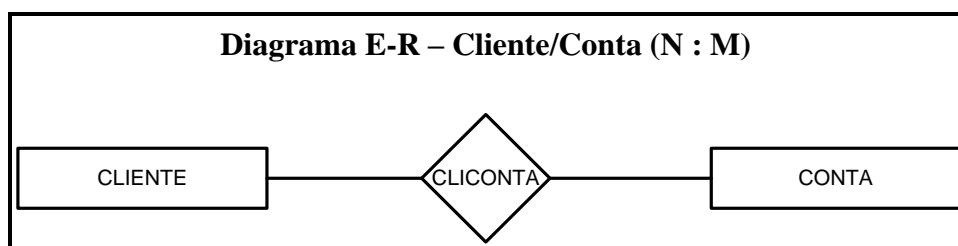


Figura 7. Exemplo 1 de Diagrama E-R (N : M)



Muitos clientes possuem uma mesma conta e uma conta pode pertencer a muitos clientes.

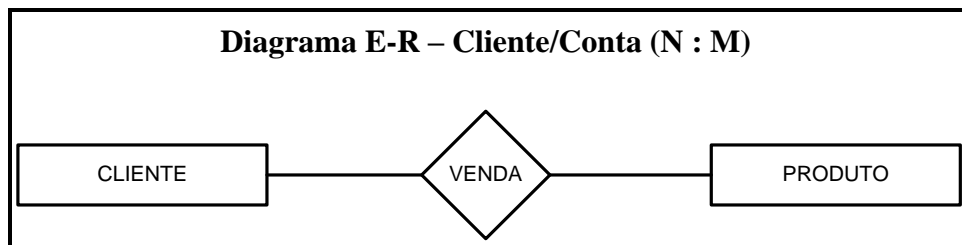


Figura 8. Exemplo 2 de Diagrama E-R (N : M)

Muitos clientes podem adquirir muitos produtos e um produto pode ser adquirido por muitos clientes.

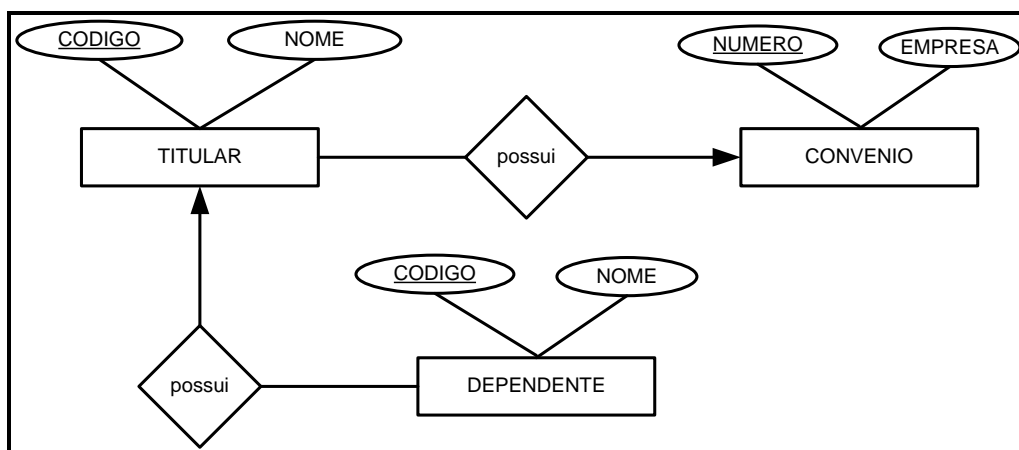


Figura 9. Exemplo de um Diagrama Entidade-Relacionamento

3.3. CONCEITO DE CHAVES

O conceito básico para identificar entidades e estabelecer relacionamentos entre essas entidades de um banco de dados é o da chave. Geralmente os bancos de dados dão possibilidade para que pelo menos dois tipos de chaves sejam utilizados: a chave primária e a chave estrangeira.



3.3.1. Chave Primária

Conforme FANDERUFF (2000, p.1), chave primária é a principal chave de acesso a uma tabela. Sua criação faz com que a tabela seja ordenada, automaticamente, por esta chave e que não seja permitida duplicidade em seu valor.

Segundo SILBERSCHATZ (1999, p.32), chave primária é um conjunto de um ou mais atributos que, tomados coletivamente, nos permitem identificar de maneira exclusiva, de maneira única, uma entidade em um conjunto de entidades.

Ainda conforme SILBERSCHATZ (1999, p.33), uma chave primária é uma propriedade do conjunto de entidades e não de uma entidade individual. Quaisquer duas ou mais entidades individuais em um conjunto de entidades não podem ter, simultaneamente, mesmos valores em seus atributos chave primária.

Uma chave primária é uma coluna ou uma combinação de colunas cujos valores distinguem uma linha das demais dentro de uma tabela (HEUSER, 2000, p.79).

CÓDIGO	DESCRIÇÃO	ESTOQUE	VALOR
0001	Arroz	25	5,95
0002	Feijão	30	1,85
0003	Batata	15	0,95
0004	Cebola	10	0,84

↑
CHAVE PRIMÁRIA

Figura 6. Uso de Chave Primária – Conjunto de Entidades PRODUTO

De acordo com a Figura 6, nota-se que cada uma das entidades, do conjunto de entidades PRODUTO, possui o valor do atributo CODIGO diferente dos demais valores das outras entidades, e isso possibilita a identificação unívoca (exclusiva) de cada uma das entidades, e evita a duplicidade. Portanto, o atributo CODIGO, ilustrado na Figura 6, exemplifica o uso de chave primária.

Os bancos de dados garantem a exclusividade de um atributo, a não duplicação de seus valores por meio da definição de uma restrição (CONSTRAINT) com o uso da cláusula PRIMARY KEY, utilizado no momento da criação da estrutura de um conjunto de entidades, como descrito no *script* da Figura 7.



```
CREATE TABLE PRODUTO1
(CODIGO NUMBER(5),
DESCRICAO VARCHAR(50),
ESTOQUE NUMBER(8,2),
VALOR NUMBER(12,2),
CONSTRAINT PRODUTO_CODIGO_PK PRIMARY KEY (CODIGO));
```

Figura 7. Exemplo de Criação de Chave Primária

Existem casos em que a escolha de apenas um atributo não é suficiente para determinar a exclusividade de uma entidade. Neste caso, é necessária a escolha de mais de um atributo para formar a chave primária, que é mais conhecida por chave primária composta. A Figura 8 ilustra o uso de chave primária composta, em que o atributo COD_TITULAR juntamente com o atributo COD_FILHO compõem (formam) a chave primária do conjunto de entidades FILHO.

COD_TITULAR	COD_FILHO	NOME	NASCIMENTO
0001	01	FERNANDA	30/03/1981
0001	02	FLÁVIA	03/05/1978
0002	01	BRENO	26/08/2003
0002	02	RENAN	15/08/2000

Figura 8. Exemplo do uso de Chave Primária Composta – Conjunto de Entidades FILHO

A definição da chave primária composta possibilitará a inclusão de 'n' filhos por titular. Os valores das chaves primárias compostas "000101" e "000102" identificam que o titular "0001" possui dois filhos, o "01" (Fernanda) e o "02" (Flávia), já os valores das chaves primárias compostas "000201" e "000202", identificam que o titular "0002" possui também dois filhos, o "01" (Breno) e o "02" (Renan). Nota-se que mesmo com a existência de filhos com códigos iguais, como a Fernanda e o Breno (representados pelo COD_FILHO '01'), não ocorrerá duplicidade de entidades, já que cada um desses filhos pertence a titulares com códigos diferentes.

As chaves primárias compostas são muito utilizadas para a construção de conjuntos de entidades que servirão de movimentações, principalmente quando ocorre o mapeamento de cardinalidade "muitos para muitos".

Para a criação de uma chave primária composta, basta acrescentar na linha de criação da chave primária os atributos que formarão a chave primária composta, como ilustrado no *script* da Figura 9.



```
CREATE TABLE FILHO
(COD_TITULAR NUMBER(5),
COD_FILHO NUMBER(2),
NOME VARCHAR(60),
NASCIMENTO DATE,
CONSTRAINT FILHO_PK PRIMARY KEY (COD_TITULAR,COD_FILHO));
```

Figura 9. Exemplo de Criação de Chave Primária Composta

3.3.2. Chave Estrangeira

Conforme HEUSER (2000, p. 80), uma chave estrangeira é uma coluna ou uma combinação de colunas, cujos valores aparecem necessariamente na chave primária de uma tabela. A chave estrangeira é o mecanismo que permite a implementação de relacionamentos em um banco de dados relacional.

Uma chave estrangeira é um conjunto de atributos de um conjunto de entidades cujos valores devem obrigatoriamente corresponder a valores de alguma chave primária de algum outro conjunto de entidades (DATE, 2003, p. 235).

Com isso, garante-se a regra de Integridade Referencial, em que um banco de dados não pode conter quaisquer valores de chaves estrangeiras que não correspondam aos valores das chaves primárias, a elas relacionadas, e vice-versa.

A Figura 10 ilustra o uso de chave estrangeira, bem como de chave primária e de chave primária composta.

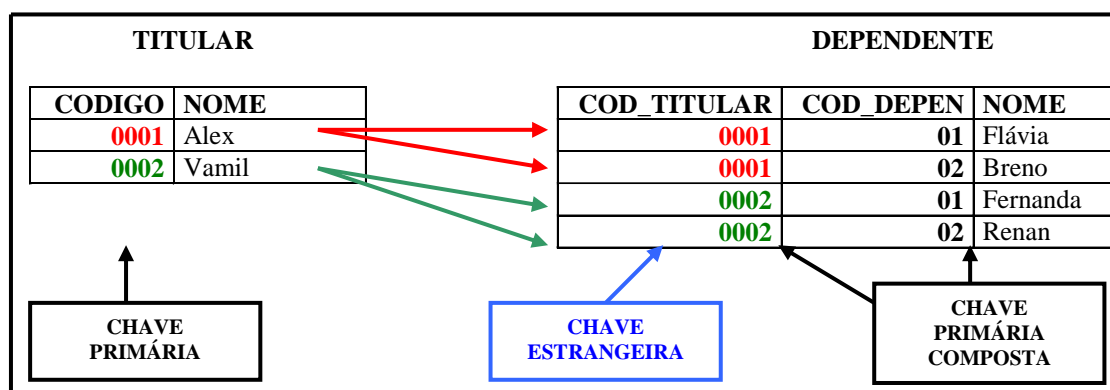


Figura 10. Uso de Chave Estrangeira – Conjunto de Entidades TITULAR relacionado com o Conjunto de Entidades DEPENDENTE



O *script* da Figura 11 ilustra a criação da chave estrangeira do conjunto de entidades DEPENDENTE, que por sua vez deverá estar referenciada a chave primária do conjunto de entidades TITULAR. A chave estrangeira deverá ser definida via criação de uma restrição (CONSTRAINT), por meio do comando FOREIGN KEY que por sua vez deverá referenciar a chave primária do conjunto de entidades a qual deverá se relacionar.

```
CREATE TABLE TITULAR
(CODIGO NUMBER(5),
NOME VARCHAR(60),
CONSTRAINT TITULAR_PK PRIMARY KEY (CODIGO));

CREATE TABLE DEPENDENTE
(COD_TITULAR NUMBER(5),
COD_DEPEN NUMBER(2),
NOME VARCHAR(60),
CONSTRAINT DEPEN_PK PRIMARY KEY (COD_TITULAR,COD_DEPEN),
CONSTRAINT DEPEN_FK FOREIGN KEY(COD_TITULAR) REFERENCES
TITULAR(CODIGO));
```

Figura 11. Exemplo de Criação de Chave Estrangeira

Com a criação da restrição “DEPEN_FK”, garante-se a Integridade Referencial, já que só poderão ser incluídas entidades, no conjunto de entidades DEPENDENTE, desde que o valor do atributo COD_TITULAR seja localizado no conjunto de entidades TITULAR, caso contrário, o dependente não poderá ser incluído. E, só se poderá eliminar uma entidade, do conjunto de entidades TITULAR, caso não seja localizado nenhuma entidade, no conjunto de entidades DEPENDENTE, que possua o valor de chave estrangeira igual ao valor da chave primária do conjunto de entidades TITULAR. Ou seja, a Integridade dos Dados é totalmente garantida e mantida quando se trabalha com chaves estrangeiras. É um recurso que deve ser totalmente explorado para que não ocorram inconsistências entre os dados.

3.4. INTEGRIDADE REFERENCIAL

Integridade Referencial refere-se a assegurar que um valor que aparece em uma tabela (relação) para um dado conjunto de atributos apareça também em certo conjunto de atributos em outra tabela (relação).

A integridade referencial surge freqüentemente ao derivarmos o esquema de banco de dados relacional construindo tabelas a partir de diagramas E-R (Entidade-Relacionamento).



A integridade referencial pode ser especificada durante a construção das estruturas de dados por meio da instrução CREATE TABLE, acrescentada das cláusulas PRIMARY KEY (chave primária), UNIQUE KEY (chave candidata), e FOREIGN KEY (chave estrangeira). (SQL-DDL).

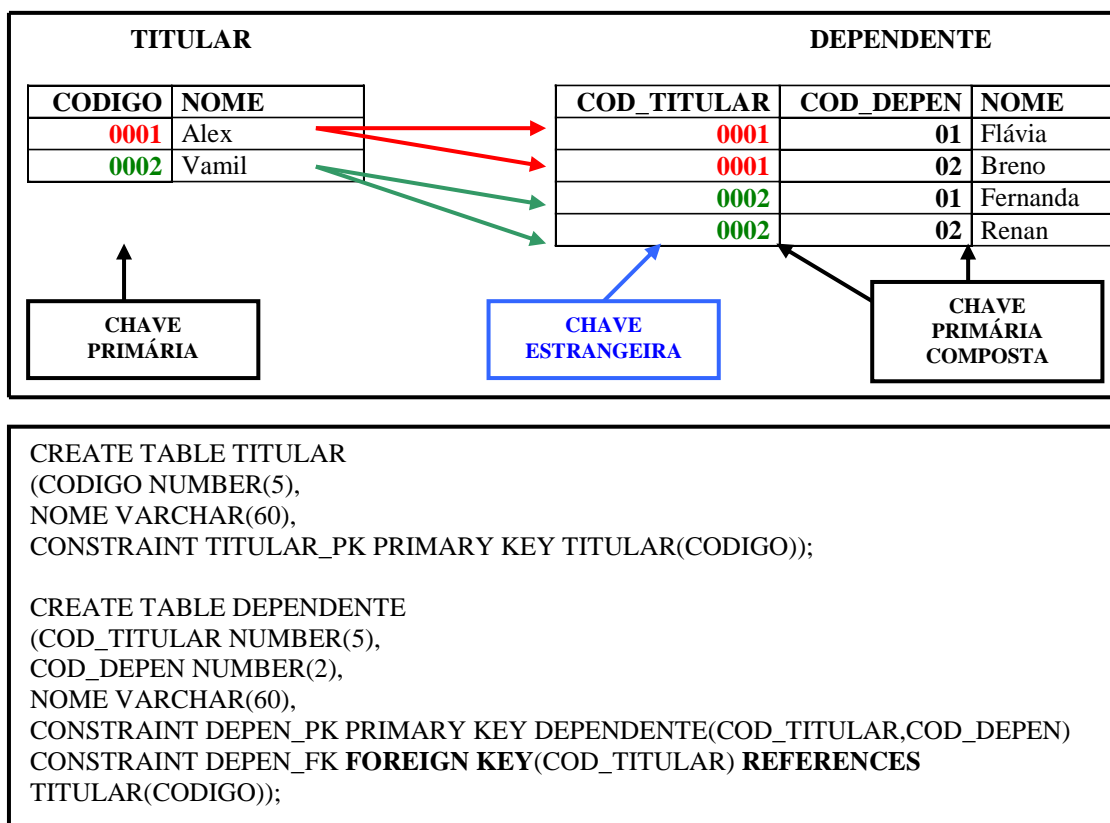


Figura 12. Ilustração de Integridade Referencial

Modificações no banco de dados, ocorridas por meio de inserções, remoções e atualizações, podem causar violações da integridade referencial, e conseqüentemente inconsistência dos dados.