



Modelo Lógico

Instituto Federal de Educação, Ciência e Tecnologia do Triângulo Mineiro

Prof. Edwar Saliba Júnior

Março de 2023



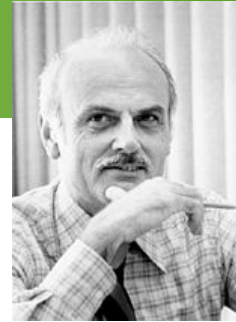
Modelo Lógico

- Dependente do SGBD (hierárquico, redes, relacional, espacial e etc.);
- trataremos somente o modelo relacional e
- é menos abstrato que o modelo conceitual (mais próximo da implementação do BD).



Banco de Dados Relacional

- Conceito abstrato que define maneiras de:
 - armazenar,
 - manipular,
 - e recuperar dados estruturados.
- Os dados estão unicamente na forma de tabelas para construir um banco de dados.
- Surgiram em meados da década de 1970 e alguns anos mais tarde as empresas passaram a utilizá-los no lugar de arquivos.
- Edgar Frank Codd foi o criador do modelo relacional (criou as 12 regras de Codd).
- A linguagem dos bancos de dados relacionais é conhecida como SQL (*Structured Query Language*).



Fonte da foto:
<http://en.wikipedia.org/wiki/Edgar_F_Codd> Acesso em: 03 fev. 2010.

As 12 Regras de Codd

- Segundo estas doze regras, discute-se a fidelidade de um banco de dados ao modelo relacional e
- raros são os bancos de dados que se enquadram em mais de dez destas doze regras propostas por Codd.



As 12 Regras de Codd

1. Toda informação num Banco de Dados Relacional (BDR) deve ser apresentada em nível lógico por valores em tabelas;
2. todo dado em um BDR deve ter a garantia de ser logicamente acessível, recorrendo-se a combinação do nome da tabela, um valor de chave e um nome de coluna;



As 12 Regras de Codd

3. deve fazer o tratamento sistemático de valores nulos (ausência de dados);
4. o dicionário de dados ativo (catálogo), deve ser baseado no modelo relacional;
5. o SGBD relacionais deve ter uma linguagem para definição, detalhamento e manipulação dos dados;



As 12 Regras de Codd

6. deve prover o tratamento de atualizações de visões;
7. deve prover o tratamento de alto nível para inserção, atualização e exclusão de dados;
8. deve possuir independência do dado físico (mudança de *hardwares* e/ou nos métodos de acesso a estes não podem interferir nos valores dos dados armazenados);



As 12 Regras de Codd

9. deve possuir independência de dados lógicos (mudanças de qualquer tipo nas estruturas das tabelas, por exemplo a inclusão de uma nova coluna, não pode interferir nos demais dados preexistentes);
10. deve possuir independência das restrições de integridade (as restrições de integridade devem ser tratadas pelo catálogo do banco e não por *softwares* externos);
11. deve possuir independência de distribuição (mesmo se o SGBD estiver em vários equipamentos ao mesmo tempo, esta distribuição deve ser transparente para o usuário) e
12. não deve haver subversão das regras de integridade ou restrições, quando se utiliza uma linguagem de baixo nível (mesmo que se tenha acesso aos dados do BD por linguagem que não seja o SQL, ainda assim, as regras de integridade e/ou restrições têm que ser respeitadas).



Conceitos

- **Domínio:**
 - define o tipo de dado e seus possíveis valores.

Exemplos:

- *nome*: cadeia de caracteres (1 a 100 caracteres);
 - *código*: número inteiro de 1 a 10.000.000 e
 - *salário*: número real (com 2 dígitos decimais);
- Um mesmo dado pode ser representado por diferentes domínios dependendo do escopo da aplicação.

Exemplo: CEP pode ser um número inteiro, ou então, um *string* de 8 ou 9 caracteres.



Domínio - Outros Exemplos

- Domínio é um conceito semântico. Exemplos:
 - *peso*, *valor* são números reais;
 - *idade*, *quantidade* são números inteiros;
 - *nome* e *município*, são caracteres (*string*, *char* ou *varchar*) e
 - *datas* são do tipo data (*date* ou *timestamp*).
- O mesmo domínio pode ser utilizado múltiplas vezes em uma relação;

Exemplo:

- relação: Peça (atributos: Código, Nome, Idade);
- três atributos diferentes com dois domínios apenas:
 - Código e Idade - inteiro e
 - Nome - caractere.



Relação

- Nome originado da matemática (teoria de conjuntos), produto cartesiano de uma lista de domínios;
- Esquema de Relação:
 - nome da relação + atributos (papel do domínio).
- Relação: esquema da Relação + tuplas = tabela;

NOMEEMP	PNOME	UNOME	SSN
Alicia	Zelaya		999887777
Jennifer	Wallace		987654321
Joyce	English		453453453



Tabela

- Todos os dados de um banco de dados relacional, são armazenados em tabelas;
- simples estrutura de linhas e colunas;
- cada linha contém um mesmo conjunto de colunas, e estas linhas devem seguir a ordem que foi especificada pelo DA (Database Administrator);
- em um banco de dados podem existir uma, dezenas, centenas ou milhares de tabelas;
- as tabelas associam-se entre si através de regras de relacionamentos;



Tabela

- em nível de modelo lógico, não nos preocupamos com a forma de ordenação das tuplas e nem com a ordem dos atributos na relação;
- o valor *NULL* é usado para indicar que o atributo não foi preenchido ou não se aplica e
- estas podem representar fatos sobre uma entidade ou um relacionamento.



Tabela

- A **relação** ou **tabela** nos SGBD's relacionais, é composta por duas partes:
 - **cabeçalho**: conjunto fixo de atributos, que são as colunas ou campos da relação e
 - **corpo**: conjunto variável de registros (tuplas), que são as linhas da relação.

Código	Nome	Função	Idade	Salário
10	Ana Maria	Analista	30	1500
20	Paulo	Programador	22	900
30	Beatriz	Analista	40	2200
40	Leonardo	Gerente	45	3500



Restrições

- Limitações nos dados em um BD relacional;
- restrições no domínio;
- Chave Primária:
 - identificação única de uma tupla,
 - uma ou mais chaves candidatas (atributos que compõem a identificação única da tupla).

Ex: NF → chaves candidatas: *número* e *série*.
- Integridade de Entidade: nenhuma chave primária pode permitir *NULL*;
- Restrições de Unicidade (Ex.: CPF).



Chave

- As tabelas se relacionam, umas com as outras, através das **chaves** e
- uma chave pode ser um, dois ou mais atributos que determinam a unicidade (identidade) de cada registro.
- **Chave Candidata**: conjunto de atributos que definem unicamente e minimamente cada tupla de uma relação.
Ex: Funcionário (Código, Nome, CPF),
Código e *CPF* são chaves candidatas.
- **Chave Primária (Primary Key)**: é uma chave escolhida, entre as chaves candidatas, para identificar a tupla na relação.
Ex: Funcionário (Código, Nome, CPF),
onde *Código* é a chave primária.



Chave

- **Chave Alternativa**: são as chaves candidatas não escolhidas para chave primária;
Ex: Funcionário (Código, Nome, CPF). *CPF* é a chave alternativa.
- **Chave Estrangeira (Foreign Key)**: conjunto de atributos de uma relação **R1**, que constituem chave primária de uma outra relação **R2**. As chaves estrangeiras funcionam como a ligação entre relações;
Ex: Funcionário (Cod-Emp, Nome, Cod-Depto)
(FK)

Departamento (Cod-Depto, Desc_Depto).
(PK)



Chave - Resumo

- **Chave Candidata**: conjunto de atributos que definem unicamente e minimamente cada registro de uma entidade;
Ex: Funcionário (Código, Nome, CPF);
Código e *CPF* são chaves candidatas.
- **Chave Primária (Primary Key)**: é uma chave escolhida entre as chaves candidatas;
Ex: *Código* foi escolhido como chave primária.
- **Chave Alternativa**: são as chaves candidatas não escolhidas para chave primária;
Ex: *CPF* é a chave alternativa.
- **Chave Estrangeira (Foreign Key)**: conjunto de atributos de uma relação que constituem chave primária de uma outra relação.

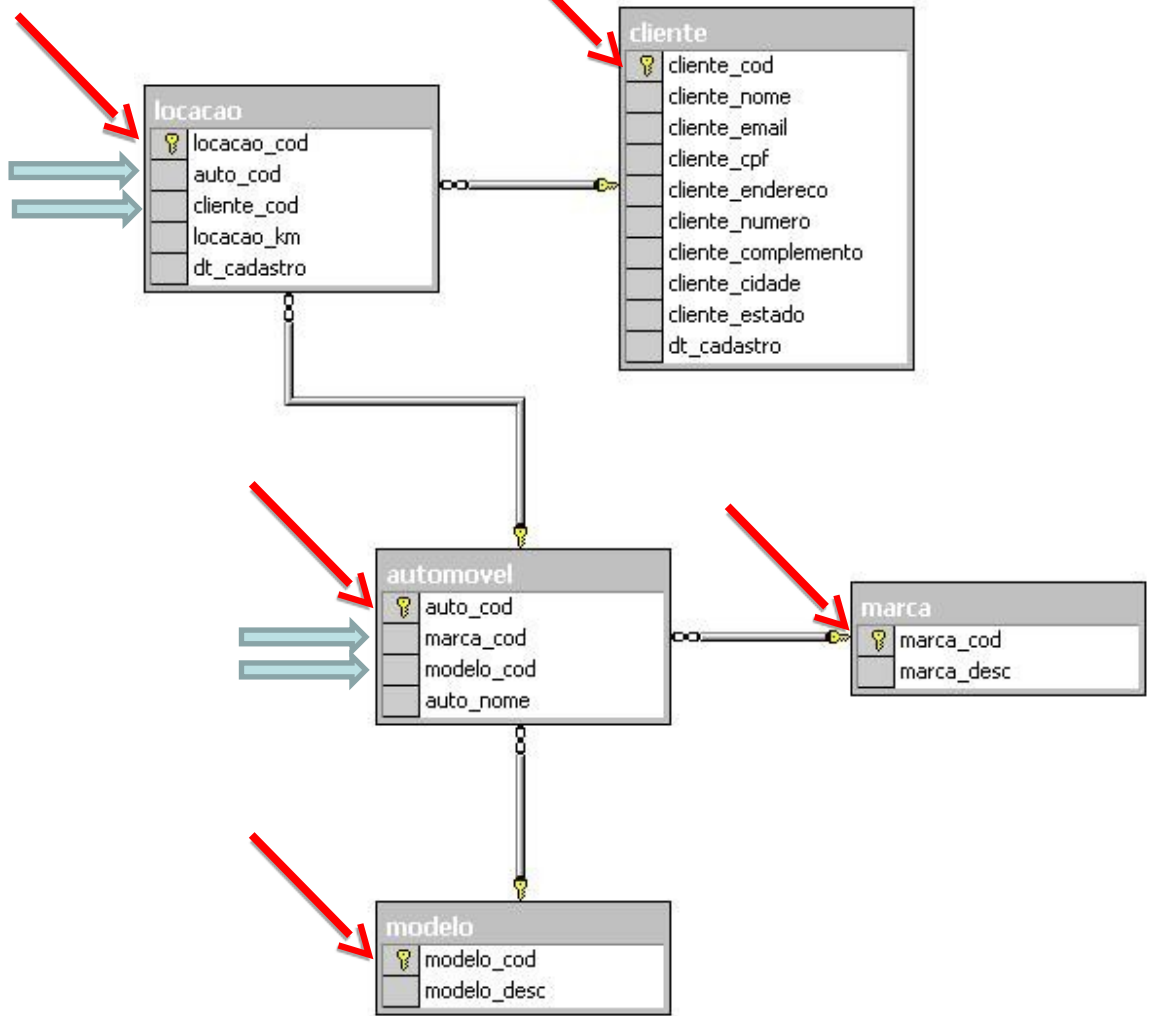


Banco de Dados I

Chave

Chave Primária (PK)

Chave Estrangeira (FK)



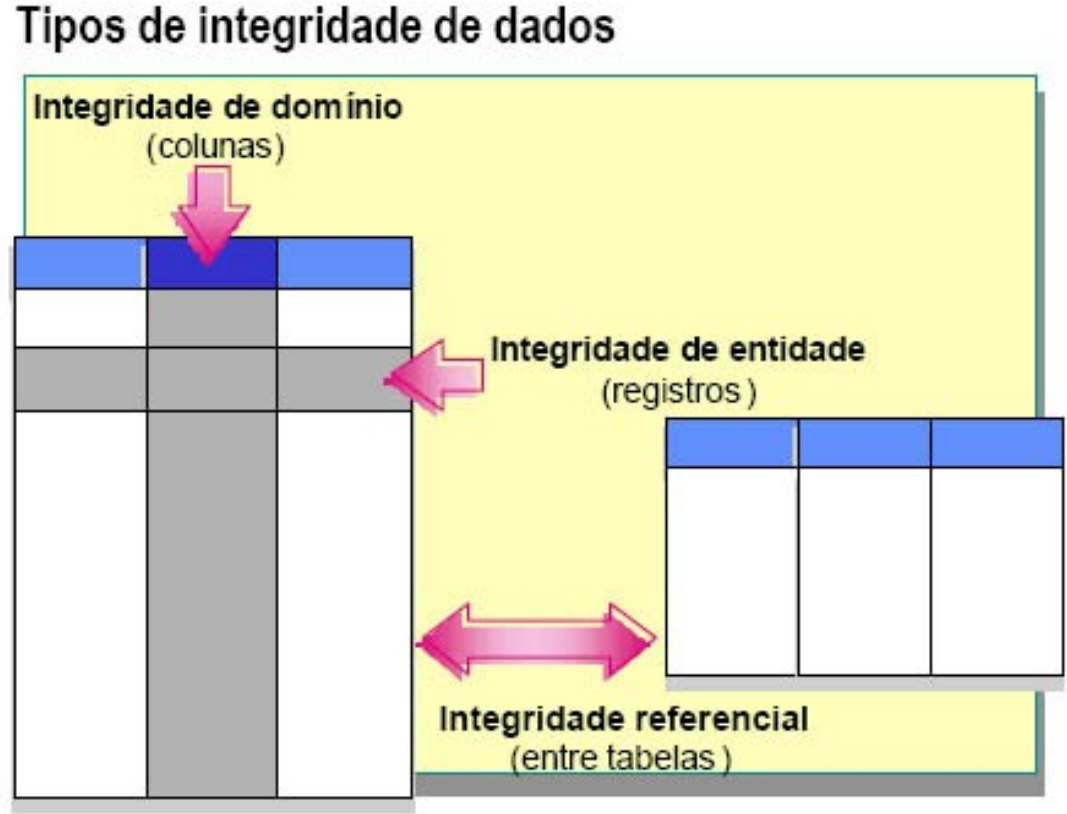


Tipos de Integridade Referencial

- ***Restrict***
 - não permite a remoção de registros que possuam relacionamentos em outras tabelas;
- ***Cascade***
 - ao se remover um registro da tabela referenciada pela chave estrangeira, os registros relacionados àquele removido serão eliminados em todas as tabelas relacionadas e
- ***Set Default e Set Null***
 - atribuem os valores *DEFAULT* ou *NULL* para as chaves estrangeiras cujos registros relacionados foram excluídos.



Tipos de Integridade de Dados





Restrições

- **Integridade Referencial:**
 - implementada através de chaves estrangeiras (FK - *Foreign Key*);
 - garante a consistência entre tuplas de duas relações ligadas conceitualmente;
 - surgem dos relacionamentos entre as entidades;
 - uma tupla de uma relação pode fazer referência à tupla de outra relação somente se esta segunda existir e
 - o relacionamento recursivo faz com que uma chave estrangeira se refira à própria relação.

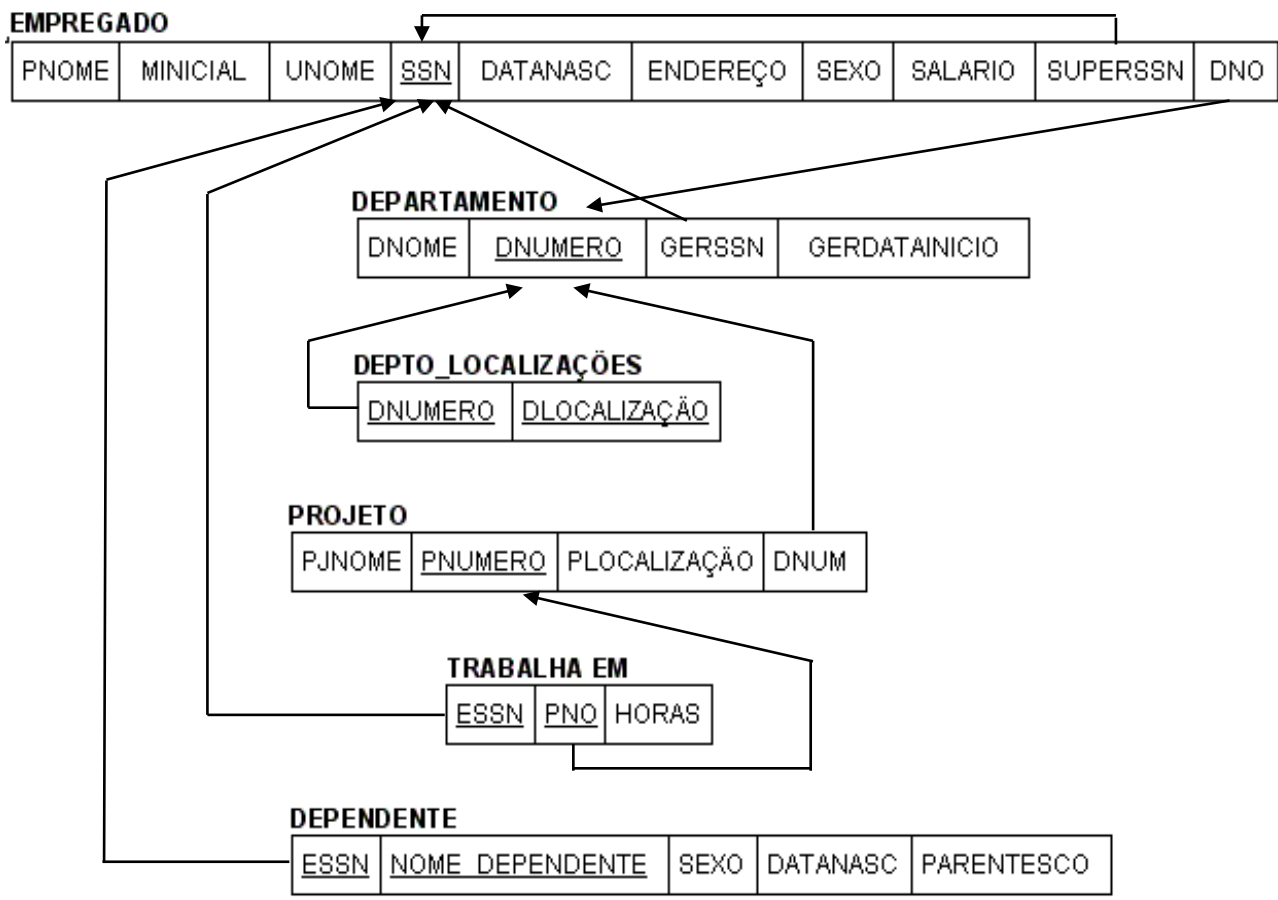


Restrições

- **Integridade Referencial - condições:**
 - o atributo A1 de uma relação R1 só pode referenciar um atributo A2 de outra relação R2, se A2 for chave primária e A1 tiver o mesmo domínio (mesmo tipo) de A2 e
 - o valor da chave estrangeira em uma tupla, ou ocorre como chave primária em alguma tupla da relação referenciada ou é nulo (*NULL*).



Exemplo





MER -> Modelo Lógico

- **Entidades Regulares:**
 - gerar uma relação (tabela) para cada entidade;
 - escolher um dos atributos chave para chave primária e
 - incluir somente atributos simples dos atributos compostos.
- **Entidades Fracas:**
 - gerar uma relação com todos os atributos da entidade fraca, e inserir como chave estrangeira, os atributos que são chave primária da entidade forte correspondente e
 - a chave primária será: a chave primária parcial da entidade fraca mais a chave primária da entidade forte.



MER -> Modelo Lógico

- **Relacionamentos Binários 1:1**
 - incluir chave estrangeira na relação gerada por qualquer das entidades que compõem o relacionamento.

- **Relacionamentos Binários 1:N**
 - uma relação para cada entidade;
 - incluir uma chave estrangeira na relação gerada pela entidade de cardinalidade N, referenciando a chave primária da relação gerada pela entidade de cardinalidade 1 e
 - no caso de relacionamento recursivo (1:N), uma relação para a entidade e uma chave estrangeira referenciando a chave primária da própria entidade.



MER -> Modelo Lógico

- **Relacionamento N:M**

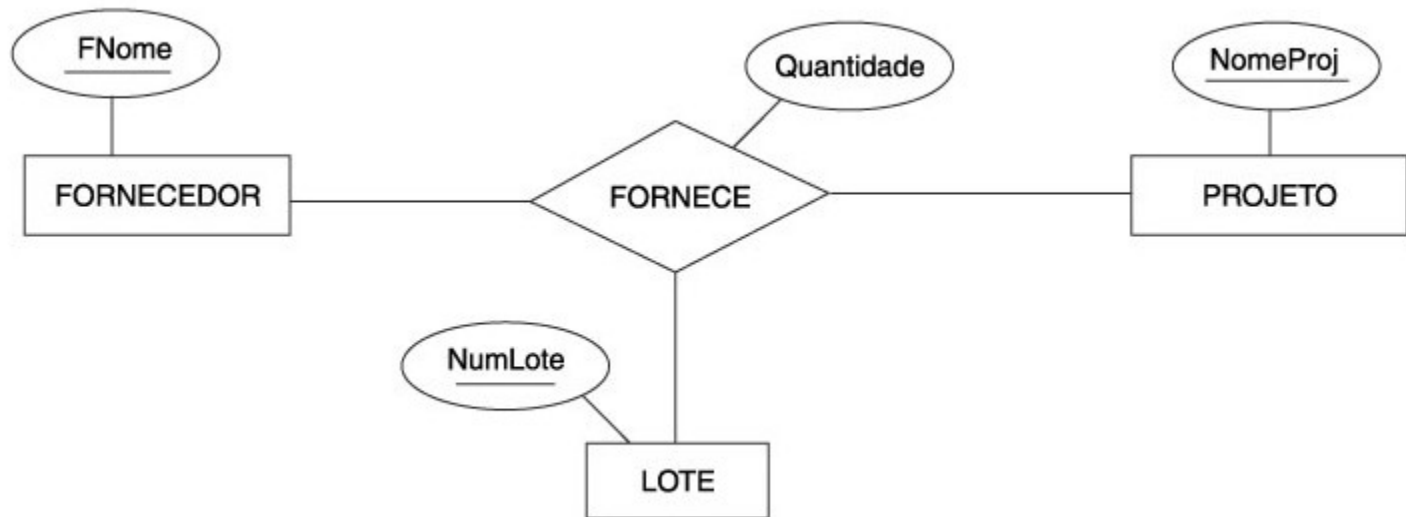
- uma relação para cada entidade e uma relação para o relacionamento;
- na relação gerada pelo relacionamento deverão existir chaves estrangeiras referenciando as chaves primárias das relações geradas pelas entidades envolvidas;
- a chave primária da relação gerada pelo relacionamento será composta pelos atributos das chaves estrangeiras e
- incluir atributos simples do relacionamento, se houver.

- **Atributos Multivalorados:**

- uma relação para cada atributo multivalorado e
- na relação gerada pela entidade, uma chave estrangeira referenciando a chave primária da nova relação.

MER -> Modelo Lógico

- **Relacionamento N-ário:**
 - uma nova relação para relacionamento com uma chave estrangeira referenciando cada uma das chaves primárias das entidades envolvidas e
 - a chave primária da nova relação será composta de todas as chaves estrangeiras;





MER -> Modelo Lógico

- **Generalização / Especialização:**

- opção 1:

- uma relação para a entidade mais genérica, e uma para cada entidade especializada (nestas, replicar a chave da relação gerada pela entidade genérica);

- opção 2:

- uma relação para cada entidade especializada; cada relação contem os atributos da entidade genérica e

- opção 3:

- uma relação única com um atributo “tipo”, e todos os atributos da entidade genérica e das entidades especializadas.



Referência Bibliográfica

ALBERT, B. E. *Slides* da disciplina Modelagem de Banco de Dados. UFRJ. Fonte: <<http://www.angelfire.com/ab7/javasenac/01mod.PDF>> Acesso em: 01 Fev. 2011.

ELMASRI, R.; NAVATHE, S. B. **Sistemas de Banco de Dados**. 4ª. Ed. São Paulo : Pearson, 2005.

ROCHA, Rogério Moraes. *Slides* da disciplina Projeto de Banco de Dados Relacional II. Faculdade INED, 2008.

SILVA, Guilherme Baião S. *Slides* da disciplina de Banco de Dados. Faculdade INED, 2006.

WIKIPEDIA. The Free Encyclopedia. Fonte <http://en.wikipedia.org/wiki/Edgar_F._Codd> Acesso em: 03 fev. 2011.

Z Aidan, Fernando. *Slides* da disciplina Banco de Dados. Faculdade Pitágoras, 2010.