

Modelagem e Projeto de Banco de Dados I

Níveis de Modelagem

Sumário

- 1 Introdução
- 2 Arquitetura ANSI-X3-SPARC
- 3 Abordagem Entidade-Relacionamento
- 4 Modelo Conceitual de Dados (MCD)
- 5 Modelo Lógico de Dados (MLD)
- 6 Modelo Físico de Dados (MFD)
- 7 Conclusões

1. Introdução

- Antes era necessário passar diretamente dos pontos (objetos) de interesse para a empresa (o negócio) para um projeto lógico de bancos de dados (planejamento da estrutura lógica de dados para o banco de dados: Relacional, Rede, Hierárquico, etc).
- O projetista de bancos de dados tinha de considerar muitas questões simultaneamente (negócio + tecnologia), o que tornava a tarefa de projeto de bancos de dados bastante difícil.

1. Introdução

- Para facilitar a tarefa de projetos de bancos de dados, surgiu então a diretriz de se obedecer um ciclo de desenvolvimento mais organizado de modelos de dados.
- Durante o ciclo de desenvolvimento, os modelos de dados passam por **níveis distintos**, cada qual com suas particularidades.
- A estratégia de utilização de diferentes níveis de projeto e representação dos modelos de dados tem origem junto ao grupo **ANSI-X3-SPARK** ainda na década de 1970.

1. Introdução

- Mais precisamente, no outono de 1971, o Comitê sobre Computador e Processamento de Informações (Comitê X3) do American National Standards Institute (ANSI) formou um grupo especial de estudos para determinar quais aspectos dos sistemas de gerenciamento de bancos de dados são candidatos adequados ao desenvolvimento de padrões.
- O grupo especial de estudos, que é chamado de Comitê de Planejamento e Requisitos de Padrões (Standards Planning and Requirements Committee - SPARC), consiste em representantes da comunidade de usuários, fabricantes de hardware e universidades.
- Surge então a Arquitetura **ANSI-X3-SPARC**

2. Arquitetura ANSI-X3-SPARC

- Esta arquitetura se divide em três níveis (ou esquemas), conhecidos como **INTERNO**, **EXTERNO** e **CONCEITUAL**, embora também sejam utilizados outros nomes.
- Cada um dos três níveis deve **ISOLAR** as características específicas inerentes.
- Sintaxes e modos de tratamento são definidos para cada um dos níveis e há grande isolamento entre eles, fazendo com que mudanças em um nível causem nenhum ou pouco impacto nos demais.

2. Arquitetura ANSI-X3-SPARC

- O nível **EXTERNO**, também conhecido como **nível lógico do usuário**, é o mais próximo dos usuários (programadores). É aquele que se ocupa do modo como os dados são vistos por **usuários (programadores) individuais**.
- O nível **INTERNO**, também conhecido como **nível de armazenamento**, é o mais próximo do meio de **armazenamento físico**, ou seja, é aquele que se ocupa do modo como os dados são fisicamente armazenados dentro do sistema.
- O nível **CONCEITUAL**, também conhecido como **nível lógico de comunidade**. É um nível intermediário de ligação (interface) entre os dois níveis anteriores.

2. Arquitetura ANSI-X3-SPARC

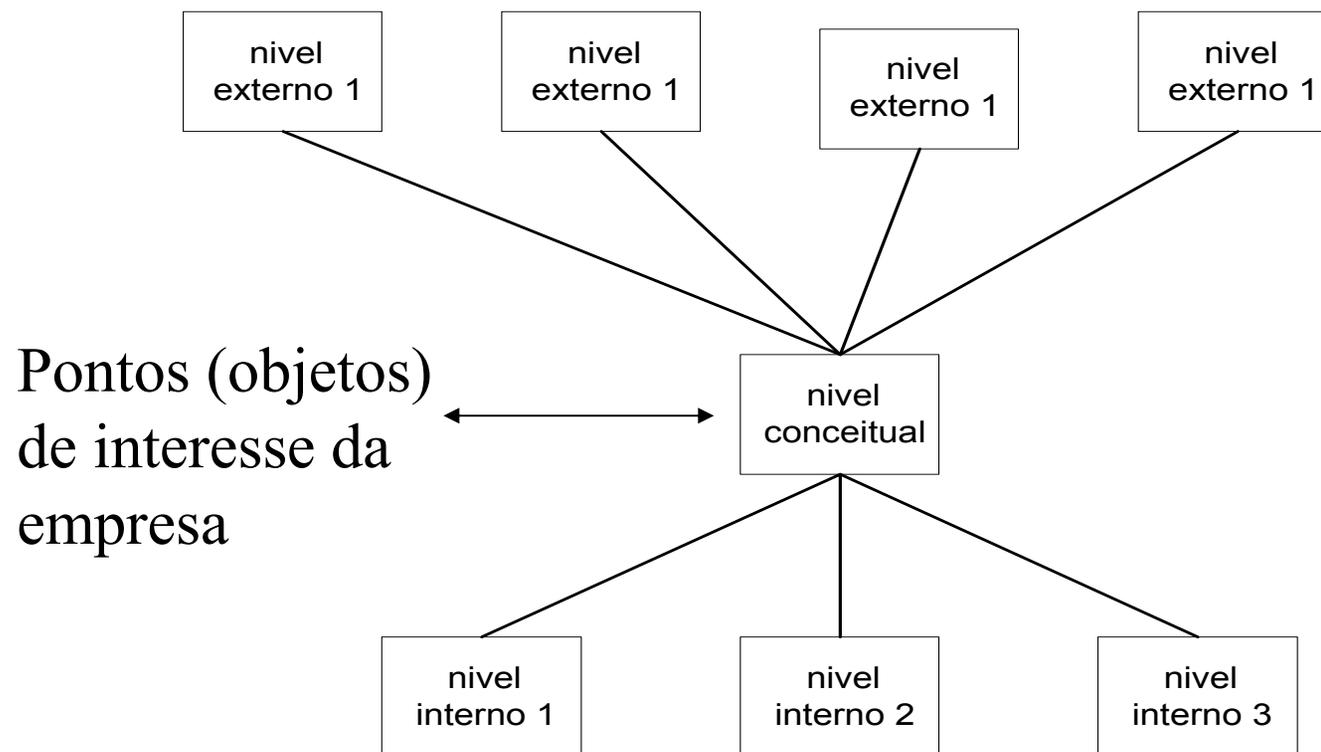
- O nível **EXTERNO** se preocupa com as percepções individuais, enquanto que o nível **CONCEITUAL** está preocupado com uma percepção conjunta da comunidade de usuários.
- A maior parte dos usuários não está interessada no banco de dados **INTEIRO**, mas apenas em alguma **PARTE** restrita.
- Assim haverá **muitas percepções distintas (EXTERNAS)**, cada qual consistindo em uma representação mais ou menos abstrata de alguma parte do banco de dados completo, e haverá exatamente uma percepção **CONCEITUAL**, consistindo em uma representação do banco de dados em sua **totalidade**.

2. Arquitetura ANSI-X3-SPARC

- Os níveis **EXTERNO** e **CONCEITUAL** são níveis de **MODELO**, enquanto que o nível **INTERNO** é mais um nível de **IMPLEMENTAÇÃO**.
- → RESUMIDAMENTE:
 - **NÍVEL EXTERNO**: se refere à organização lógica dos dados (ex: Rede, Hierárquico, Relacional, etc) ,
 - **NÍVEL CONCEITUAL**: tem um escopo mais abstrato (ex: entidades, atributos, relacionamentos, etc).
 - **NÍVEL INTERNO** se refere à organização física dos dados. É definido em termos de construções voltadas para a máquina (bits e bytes, alocação dinâmica/estática, etc).

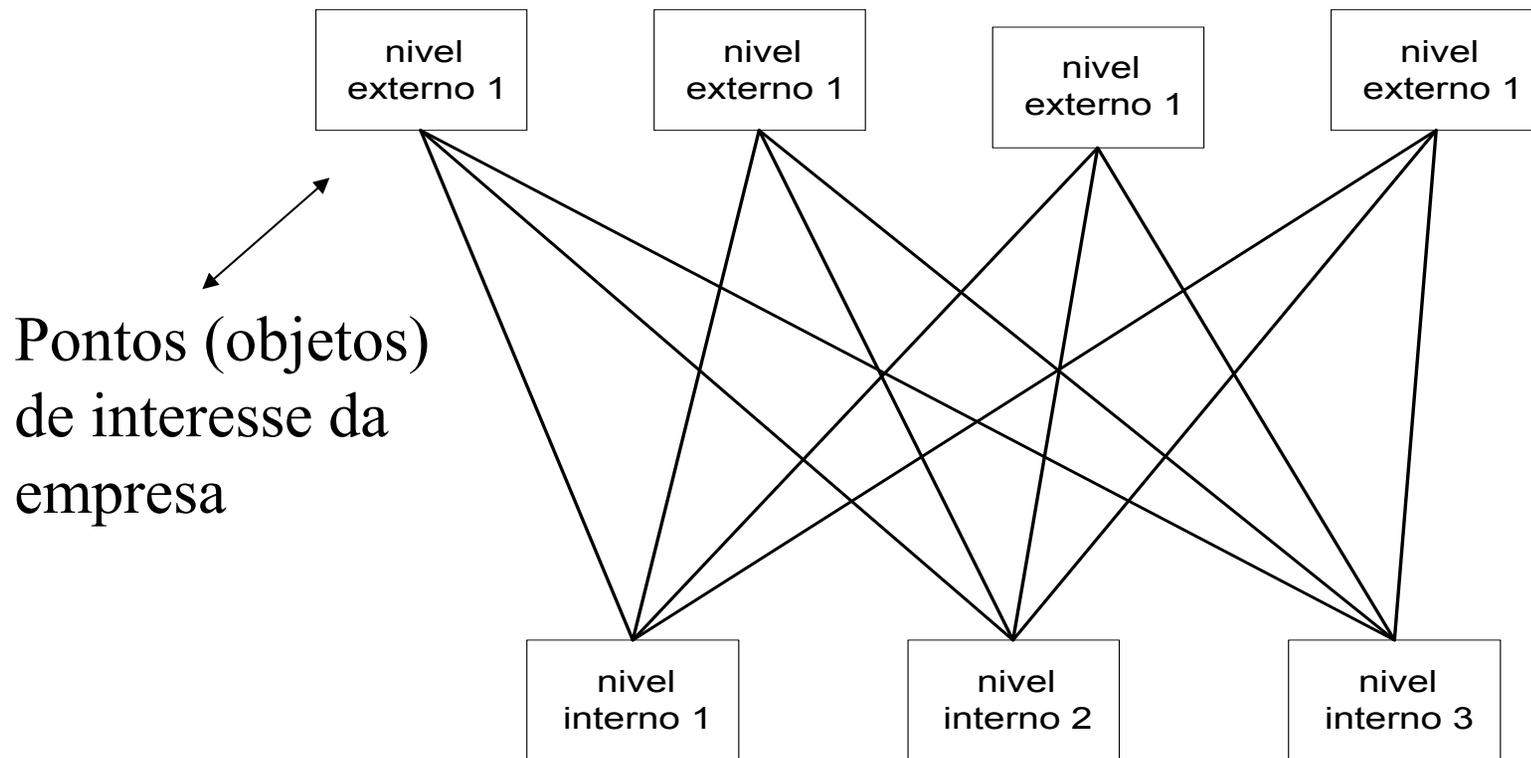
2. Arquitetura ANSI-X3-SPARC

Por exemplo, uma situação em que um modelo de nível conceitual precisasse ser definido para três diferentes modos de implementação (nível interno) convivendo com quatro diferentes visões (nível externo). Temos assim $4 + 3 = 7$ mapeamentos



2. Arquitetura ANSI-X3-SPARC

SEM o nível conceitual: note que cada uma das interligações representa uma possível combinação de características específicas de um modo de implementação e de uma visão a ser disponibilizada ao usuário. Precisamos de $4 \times 3 = 12$ mapeamentos para uma implementação flexível do banco de dados. Daí a grande importância da arquitetura ANSI-X3-SPARC.



3. Abordagem Entidade-Relacionamento

- **Em março de 1976, Peter Chen** publicou um trabalho intitulado “**The Entity-Relationship Model: Toward the unified view of data**”, o qual definia uma possível abordagem para o processo de modelagem de dados, doravante referenciada com **Abordagem E-R (Entidade-Relacionamento)**.
- **A abordagem E-R** defende, basicamente, a elaboração de um modelo que represente os objetos observados e seus relacionamentos, **independentemente** de preocupações com implementações lógicas ou físicas → **DIAGRAMA ENTIDADE-RELACIONAMENTO**
- A abordagem E-R entende que os **aspectos lógicos e físicos** devem ser agregados **a posteriori**, pois são alheios à estrutura inerente dos dados observados em um ambiente ou em um conjunto de objetos

3. Abordagem Entidade-Relacionamento

Definições **BÁSICAS** com respeito à elaboração do DIAGRAMA E-R

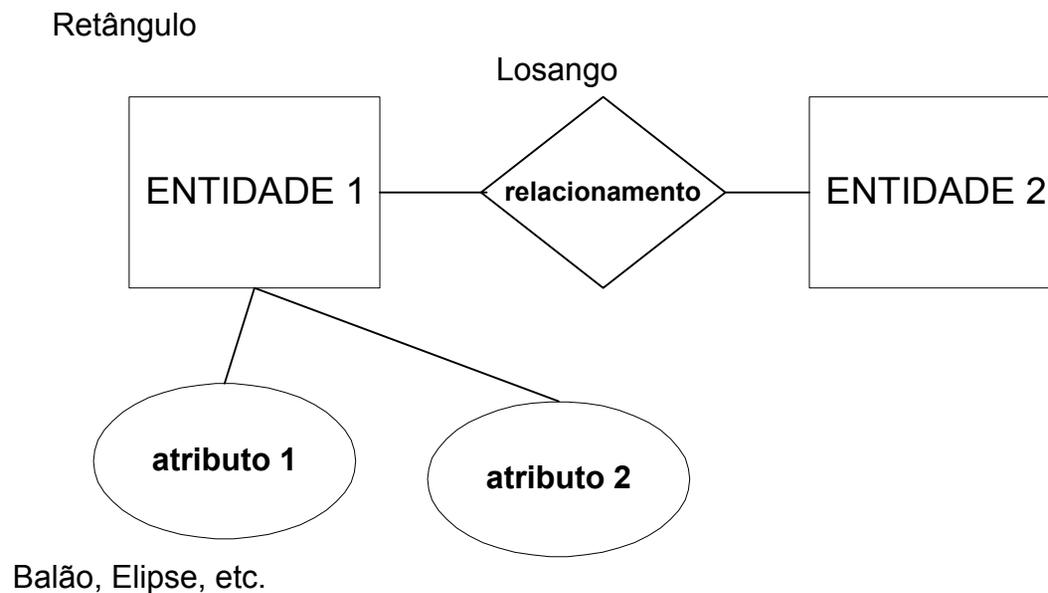
Uma **entidade** é um objeto que existe e é distinguível de outros objetos. Uma entidade pode ser um objeto com uma existência física (**entidade concreta**) - um empregado, pessoa, carro, casa em particular - ou conceitual (**entidade abstrata**) - uma companhia, um emprego, um curso universitário.

Cada entidade tem **atributos** - propriedades particulares que a descrevem. Por exemplo, uma entidade **Aluno** pode ser descrito pelos atributos **"nome"**, **"matrícula"** e **"curso"**.

Um **relacionamento** é uma associação entre duas ou mais entidades. Por exemplo, podemos definir um relacionamento que associe um aluno particular com uma certa disciplina; isto especifica que ele é um aluno da universidade, cursando aquela disciplina.

3. Abordagem Entidade-Relacionamento

Representação de DIAGRAMA E-R (Peter Chen):



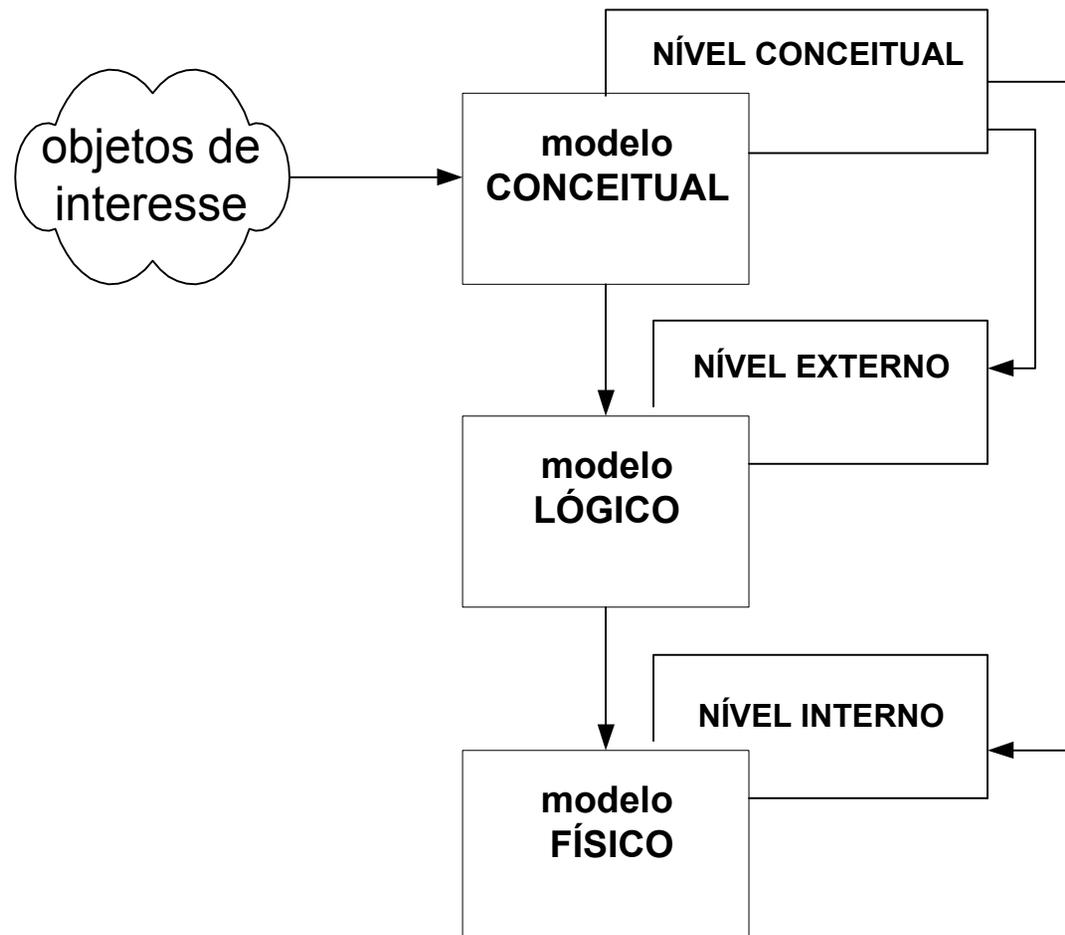
3. Abordagem Entidade-Relacionamento

- A **abordagem E-R** tem aderência aos padrões definidos na arquitetura de três níveis **ANSI-X3-SPARK** e, através de sua aplicação, nos levará, durante o ciclo de desenvolvimento, à convivência com **três** tipos distintos de **modelos de dados**:

- **MODELO CONCEITUAL DE DADOS (MCD)**
- **MODELO LÓGICO DE DADOS (MLD)**
- **MODELO FÍSICO DE DADOS (MFD)**

3. Abordagem Entidade-Relacionamento

- Integração da abordagem E-R e a proposta ANSI-X3-SPARK:



3. Abordagem Entidade-Relacionamento

- A partir da figura do **slide anterior** temos:
 - Partindo-se de um **mundo observado**, composto de seus objetos e relacionamentos, podemos definir um modelo independente de tecnologia (relacional, rede, hierárquica, OO, etc) → **MODELO CONCEITUAL**
 - A partir do **MODELO CONCEITUAL** podemos então derivar um **MODELO LÓGICO** para em seguida obter o **MODELO FÍSICO** (de implementação).
 - A cada um dos níveis de modelagem serão associadas **técnicas de representação gráfica** e métodos de especificação. Este tipo de especialização para cada um dos níveis fará com que sejam necessários conhecimentos específicos para o entendimento e manipulação dos elementos envolvidos em seus escopos.

4. Modelo Conceitual de Dados (MCD)

- É aquele em que os objetos, suas características e relacionamentos têm a representação **fiel ao ambiente observado**, independentemente de limitações quaisquer impostas por tecnologias, técnicas de implementação ou dispositivos físicos.
- Nesse modelo devemos representar os conceitos e características observados em um dado ambiente, **voltando-se simplesmente ao aspecto conceitual**
- Esse modelo deve ser o modelo a ser utilizado para o nível de **conversação, entendimento, transmissão, validação de conceitos, mapeamento do ambiente, etc.**
- Devem ser **ignoradas** quaisquer particularidades de **implementação**, bem como desconsiderada qualquer preocupação com qual será o modo de implementação futura. O modelo conceitual deve ser **imutável**.

4. Modelo Conceitual de Dados (MCD)

- A **estabilidade é o diferencial** na utilização do modelo conceitual de dados. Devemos derivar diferentes estruturas de implementação a partir de um **mesmo modelo conceitual**.
- Assim, a partir de um **mesmo MCD** podemos derivar um modelo para implementação, por exemplo, em **SGBD relacional ou OO**.
- Devemos estar sempre atentos para **NÃO** construirmos nossos MCD como modelos **LÓGICOS** de dados. Apesar de isso ser possível, devemos, mesmo que não nos pareça necessário, **construir os modelos CONCEITUAIS**, sob pena de nos restringirmos no processo de modelagem a recursos disponibilizados por apenas esta ou aquela tecnologia de implementação.
- No **MCD** devemos nos concentrar puramente no **aspecto conceitual**, obtendo mais e melhores detalhes sobre os objetos que nos interessam do ambiente observado naquele instante .

5. Modelo Lógico de Dados (MLD)

- O **MLD** é aquele em que os objetos, suas características e relacionamentos têm a representação **de acordo** com as **regras de implementação e limitantes impostos por algum tipo de tecnologia**.
- Essa **representação**, contudo, é **independente dos dispositivos ou meios de armazenamento físico** das estruturas de dados por ela definidas.
- O **MLD** deve ser o modelo elaborado **respeitando-se e utilizando-se conceitos** tais como chaves, arrays, normalização, ponteiros, entre outros. No **MLD** dizemos o que utilizar/usar (p.ex., chaves, ponteiros, etc) mas **não como implementar fisicamente**.
- A obtenção do **MLD** se dá pela **aplicação de regras de derivação** sobre um MCD já construído.
- **Este será o tipo de abordagem** de obtenção do **MLD** que **defenderemos**, baseado no fato de que, como pré-requisito, já estabelecemos a necessidade de elaboração de um **MCD**.

6. Modelo Físico de Dados (MFD)

- O **MFD** é aquele em que a representação dos objetos é feita sob o foco do **nível físico de implementação**. O conhecimento do modo físico de implementação das estruturas de dados é o ponto básico desse tipo de modelo.
- No **MFD** nos preocupamos **como implementar fisicamente** aquilo que foi estabelecido no **MLD**.
- **Cada** diferente **SGBD** poderá definir um diferente modo de **implementação física** das características e recursos necessários para o armazenamento e manipulação das estruturas de dados.
- **Também**, em alguns casos, **um mesmo SGBD**, em diferentes ambientes de **sistema operacional (S.O)**, poderá ter diferentes métodos de armazenamento e manuseio de estruturas de dados. Em cada diferente ambiente de S.O podem existir, por exemplo, diferentes estruturas de armazenamento, endereçamento, acesso e alocação física.

6. Modelo Físico de Dados (MFD)

- Assim, note que um mesmo **MODELO LÓGICO DE DADOS (MLD)** pode ser mapeado de diferentes modos, dependendo do sistema operacional considerado.
- Em nosso curso, **NÃO** trataremos do **MODELO FÍSICO DE DADOS (MFD)**.
- Nossos estudos voltam-se para o **MODELO CONCEITUAL DE DADOS (MCD)** e o **MODELO LÓGICO DE DADOS (MLD)**.

7. Conclusões

- Nas próximas notas de aula detalharemos a abordagem Entidade-Relacionamento. Aguarde! 😊

FIM