

Resumo Aulas Base de Dados

Contents

| | | |
|----------|--|----------|
| 1 | Definição do Sistema | 2 |
| 1.1 | O Problema | 2 |
| 1.2 | Fundamentação | 2 |
| 1.3 | Análise de Viabilidade | 2 |
| 1.4 | Levantamento de Requisitos | 2 |
| 2 | Desenho Conceptual | 3 |
| 2.1 | Etapas | 3 |
| 2.2 | Entidades | 3 |
| 2.3 | Relacionamentos | 3 |
| 2.4 | Atributos | 4 |
| 2.4.1 | Domínio dos atributos | 5 |
| 2.4.2 | Atributos Chave | 5 |
| 3 | Exemplo de modelo | 6 |
| 4 | Desenho Lógico | 6 |
| 4.1 | Construção dos Modelos de Dados Lógicos locais | 7 |
| 4.2 | Atributos Multivalor | 9 |
| 4.3 | Mapeamento Conceptual-Lógico | 10 |
| 4.3.1 | Relacionamentos 1:N | 10 |
| 4.3.2 | Relacionamentos M:N | 10 |
| 4.3.3 | Relacionamentos com atributos | 11 |
| 4.3.4 | Relacionamentos recursivos | 11 |
| 4.4 | Validação do Modelo | 12 |
| 4.5 | Análise de Interrogações | 12 |
| 4.5.1 | Exemplos de interrogações | 12 |
| 4.6 | Análise de Transações | 13 |
| 4.7 | Tipos de Restrições de Integridade | 13 |
| 4.8 | Validar o Modelo de Dados Global | 13 |
| 4.9 | Analisar a Evolução da Base de Dados | 13 |
| 4.10 | Versão Final do Modelo Lógico | 14 |

1 Definição do Sistema

1.1 O Problema

É importante que antes de se começar a realizar um processo de levantamento de requisitos se conheça de forma detalhada e suportada o porquê de levar a cabo um processo de construção de base de dados. A contextualização, fundamentação e realização de uma análise de viabilidade contribuem para a compreensão do problema por toda a equipa de desenvolvimento, bem como contribui para o levantamento de requisitos.

1.2 Fundamentação

A fundamentação do processo de desenvolvimento de uma base de dados deve descrever o porquê da necessidade da base de dados. É então pretendido que se apresente de uma forma breve, os sistema e processos que esta vai suportar. Deve também elucidar como é que a base de dados se integrará na estrutura computacional da organização.

1.3 Análise de Viabilidade

Deve ser sustentada por um pequeno estudo, cujo principal objetivo é responder às seguintes questões:

- Principais objetivos a atingir com a construção da base de dados?
- Será que a base de dados poderá ser construída de acordo com o calendário estabelecido e com os recursos disponíveis?
- Os sistemas cuja operacionalidade será suportada pela base de dados estão preparados adequadamente?
- (...)

1.4 Levantamento de Requisitos

Neste processo procura-se identificar e caracterizar os elementos de dados mais relevantes para o problema em causa, que precisam por um qualquer motivo de serem guardadas na futura base de dados.

Deve-se ter o cuidado de organizar de acordo com as diversas vertentes de implementação:

- Administração - acesso, monitorização, segurança, etc.
- Exploração - queries, relatórios, procedimentos, etc.
- Descrição - objetos de dados, domínios, restrições, etc.

2 Desenho Conceptual

Em termos de documentação deve-se ter o cuidado de enumerar tudo aquilo que é necessário para o esquema de base de dados e escrever todas as peças elementares. Conforme se vai levantando os requisitos, vai-se fazendo o esquema conceptual e identificando os diversos componentes.

Tem de ser claro, fácil de interpretar e não ambíguo.

2.1 Etapas

O processo de desenvolvimento do modelo de dados conceptual pode incluir as seguintes fases:

- Construção do modelo de dados conceptual local para cada vista de utilizador;
- Identificação dos tipos de entidades;
- Identificação dos tipos de relacionamentos
- Identificação e associação dos atributos com tipos de entidades ou relacionamentos;
- Identificação dos domínios dos atributos;
- Identificação dos atributos chave;
- Detalhe ou generalização dos tipos de entidades;
- Desenho de diagramas ER;
- Revisão do modelo de dados com o utilizador.

2.2 Entidades

Uma entidade é um objeto ou um conceito que está claramente identificado numa organização como tendo uma existência independente.

ex: clientes, fornecedores, produtos, máquinas, produtores, etc...

2.3 Relacionamentos

A identificação dos relacionamentos pode ser feita com base nos requisitos dos utilizadores ou nas como são referidas eventuais ligações que existem entre as entidades.

Na maioria dos casos são binários (envolvem duas entidades) mas poderão existir mais complexos, envolvendo três ou mais entidades ou relacionamentos recursivos que envolvem uma única entidade.

Na identificação de um relacionamento devemos ter em conta os seguintes aspetos:

- Determinar a cardinalidade e restrições de participação dos tipos de relacionamentos;
- Documentar os tipos de relacionamentos;
- Utilizar diagramas ER (Entidade-Relacionamento) no processo de modelação.

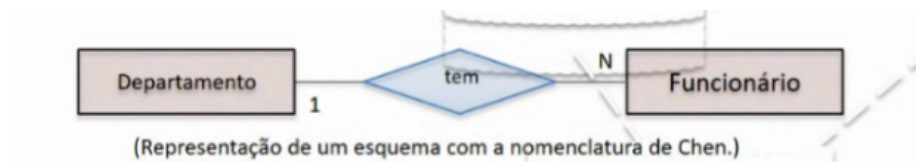


Figure 1: Exemplo de um relacionamento.

A sua documentação deverá incluir os seguintes elementos:

- Nome das entidades envolvidas;
- Designação do relacionamento;
- Cardinalidade (1:1, 1:N, N:M);
- Participação;

2.4 Atributos

O processo de identificação de um atributo considera a atribuição de uma designação que deverá ser esclarecedora para o utilizador acerca do seu significado ou conteúdo.

Tenta identificar nomes, propriedades ou características das entidades ou relacionamentos.

Têm diferentes tipos:

- Simples: que são peças elementar de caracterização de uma entidade ou relacionamento;
- Compostos: são constituídos por vários atributos simples. Quando não existe a necessidade de aceder separadamente a cada um dos atributos simples que compõem um atributo, o atributo composto deverá ser considerado como um atributo simples.
- Derivados ou calculados: são atributos cujo valor depende de um processo de cálculo qualquer. Não são introduzidos diretamente pelo utilizador.

2.4.1 Domínio dos atributos

O domínio de um atributo é um conjunto de valores dos quais um ou mais atributos assumem os seus valores.

Um modelo de dados completo deve apresentar a especificação dos domínios para todos os atributos e inclui normalmente:

- O conjunto de valores permitidos para um atributo;
- Tamanhos e formatos dos campos dos atributos.

| Nome da Entidade | Atributo | Descrição | Tipo de Dados & Comprimento/Domínio | Nulo | Tipo de Atributo | Valor por Defeito |
|------------------|------------------|-----------------------------------|-------------------------------------|------|------------------|-------------------|
| Cliente | <u>IdCliente</u> | Identificador do Cliente | INT | Não | Simples | Auto Incrementado |
| | Nome | Nome do Cliente | VARCHAR(20) | Não | Simples | - |
| | Cidade | Cidade onde o cliente vive | VARCHAR(15) | Não | Simples | - |
| | Email | Endereço eletrónico de um cliente | VARCHAR(45) | Não | Simples | - |
| | Telefone | Número Telefónico | VARCHAR(15) | Não | Simples | - |
| | Password | Chave-secreta do Cliente | VARCHAR(10) | Não | Simples | - |

Figure 2: Exemplo de atributos

2.4.2 Atributos Chave

Uma chave candidata é um atributo, ou um mínimo conjunto, de uma entidade que identifica de forma única uma instância dessa entidade. Podemos identificar mais do que uma chave candidata, mas devemos escolher apenas uma chave primária. As restantes chaves candidatas são normalmente designadas por chaves alternativas ou secundárias.

A chave candidata tem características próprias: não permite valores repetidos nem nulos.

Um Exemplo Concreto

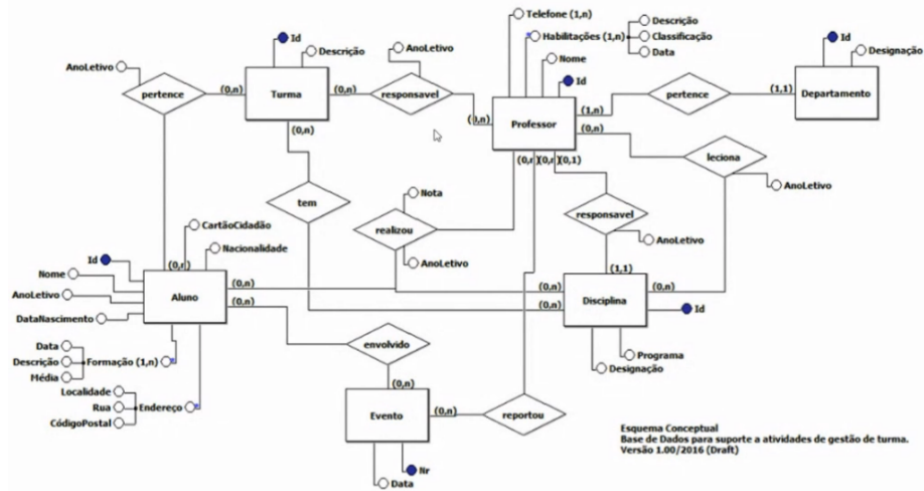


Figure 3: Exemplo concreto de um modelo.

3 Exemplo de modelo

Na aula de dia 22/10, o professor apresentou um exemplo de um modelo conceitual com introdução, definição do sistema, etc etc etc. Podemos utilizar como exemplo para o relatório do TP :)) .

4 Desenho Lógico

O projeto lógico de uma base de dados é o processo de construção de um dado modelo de informação baseado num modelo de dados específico, mas que é independente de qualquer SGBD ou de outras considerações físicas.

As principais etapas do projeto lógico de uma base de dados são:

- Construção e validação do modelo de dados lógico local (ou parcial) correspondente a cada vista de utilizador;
- Construção e validação do modelo de dados lógico global.

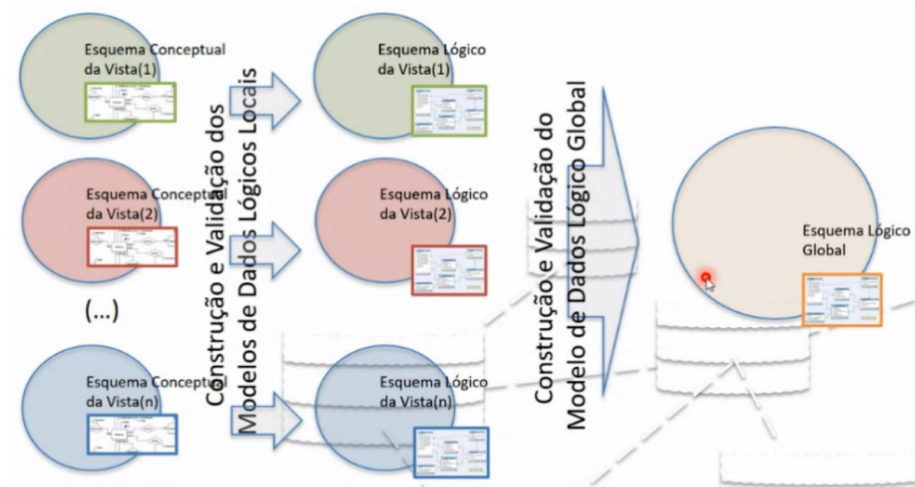


Figure 4: Etapas projeto lógico.

4.1 Construção dos Modelos de Dados Lógicos locais

Em **primeiro lugar**, faz-se a conversão dos modelos conceptuais locais para os correspondentes modelos lógicos, realizando as seguintes tarefas:

- Derivação de relações do modelo de dados local;
- Validação do modelo através da normalização;
- Validação do modelo com as transações do utilizador;
- Desenho do modelo lógico;
- Definição de restrições de integridade;
- Revisão do modelo de dados local com o utilizador.

O principal objetivo desta etapa é o refinamento do modelo de dados conceptual local, ao sentido de remover as suas características indesejáveis e fazer o seu mapeamento para um modelo de dados lógico.

Nesta altura do projeto, tem-se um modelo de dados conceptual para cada uma dada vista de utilização de um utilizador, no caso de termos optado por uma abordagem multivista, obviamente.

O Elemento de Dados

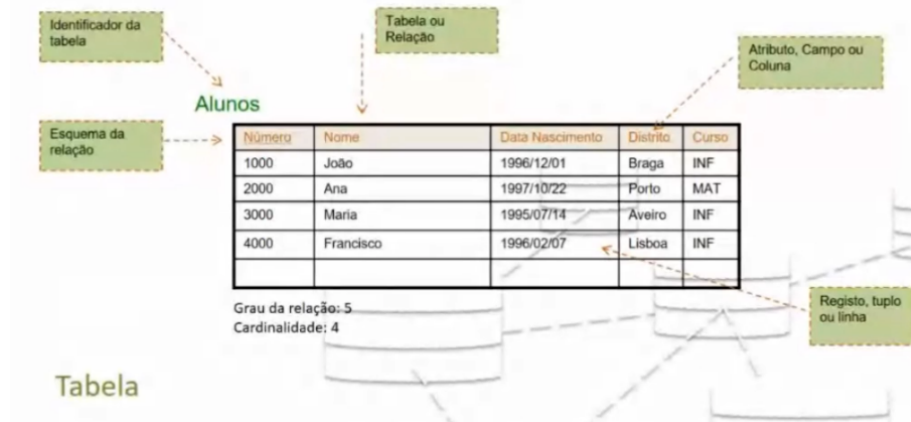


Figure 5: Elemento de dados

Um Esquema Relacional

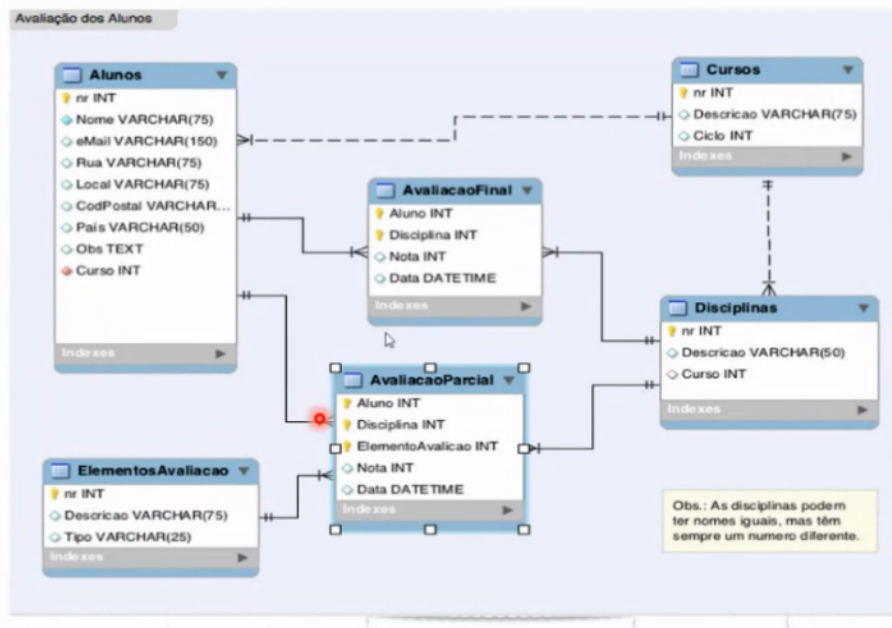


Figure 6: Esquema Relacional

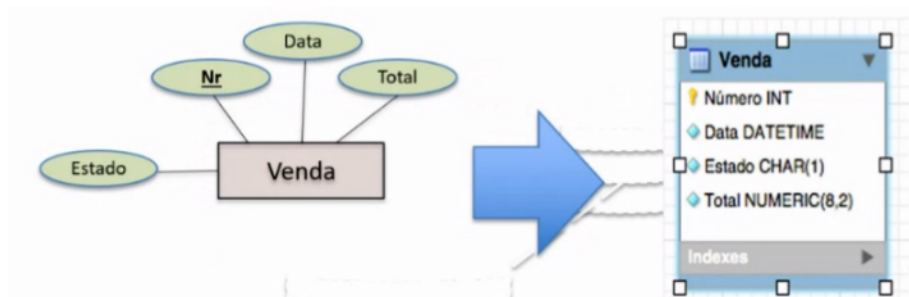


Figure 7: Mapeamento de entidades ex.1

4.2 Atributos Multivalor

Um atributo multivalor acolhe vários valores para uma única entidade. Se um atributo multivalor estiver representado no modelo de dados, devemos fazer a decomposição no sentido de identificar uma entidade correspondente.



Figure 8: Mapeamento de entidades ex.2

4.3 Mapeamento Conceptual-Lógico

Os relacionamentos definidos no modelo conceptual são mapeados através da realização de operações de transformação estabelecidas de acordo com a sua cardinalidade e atributos associados. De referir:

- "Remove" relacionamentos M:N, com ou sem atributos;
- "Remove" relacionamentos complexos;
- "Remove" relacionamentos recursivos.

Depois do mapeamento dos relacionamentos entramos na fase final de transformação, na qual devemos ver como:

- Reexaminar os relacionamentos 1:1;
- Remover relacionamentos redundantes.

4.3.1 Relacionamentos 1:N

Este tipo de relacionamentos é aquele que mais encontramos em modelos conceptuais de base de dados.

O atributo que sustenta o relacionamento é designado por **chave estrangeira** ou **externa**.

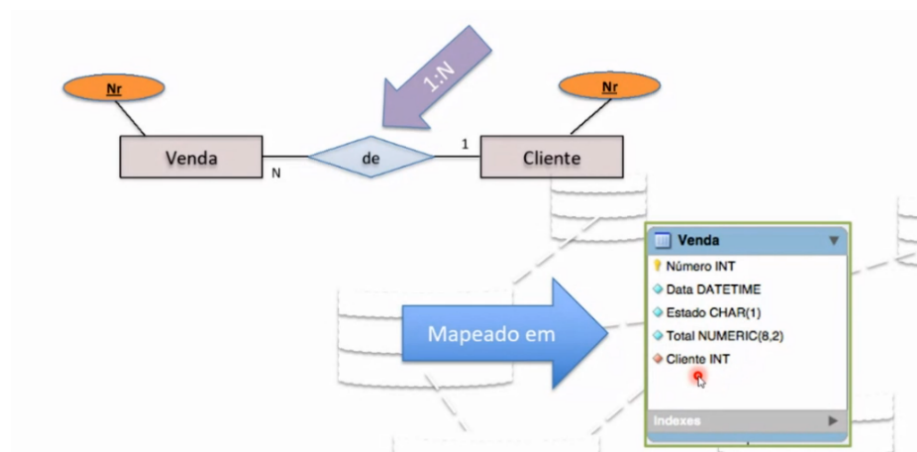


Figure 9: Relacionamentos 1:N

4.3.2 Relacionamentos M:N

Se no modelo conceptual estiver representado um relacionamento N:M, deverá-se decompô-lo no sentido de identificar uma "entidade" intermédia. O relacionamento será então substituído por dois relacionamentos 1:N para a nova "entidade".

4.3.3 Relacionamentos com atributos

Se num modelo conceptual existirem relacionamentos com atributos, então dever-se-á fazer a decomposição do relacionamento para identificar uma "identidade" intermédia. A decomposição num relacionamento N:M originará uma "entidade" fraca e dois relacionamentos 1:N.

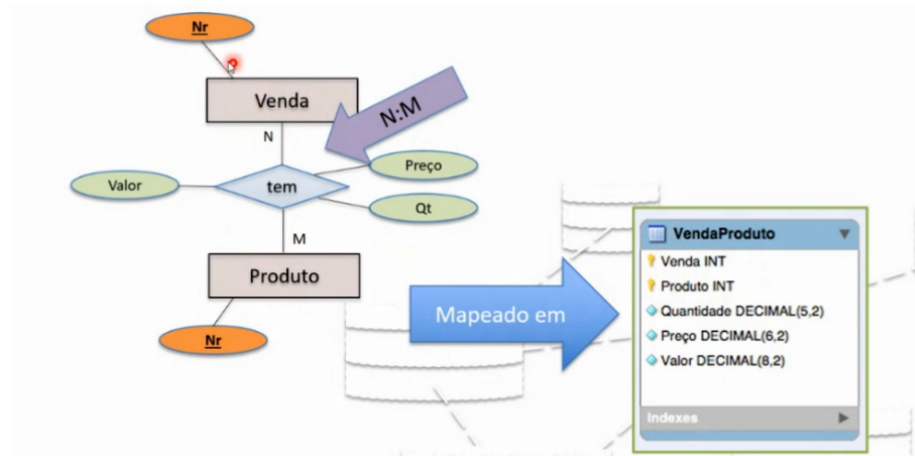


Figure 10: Relacionamentos M:N

4.3.4 Relacionamentos recursivos

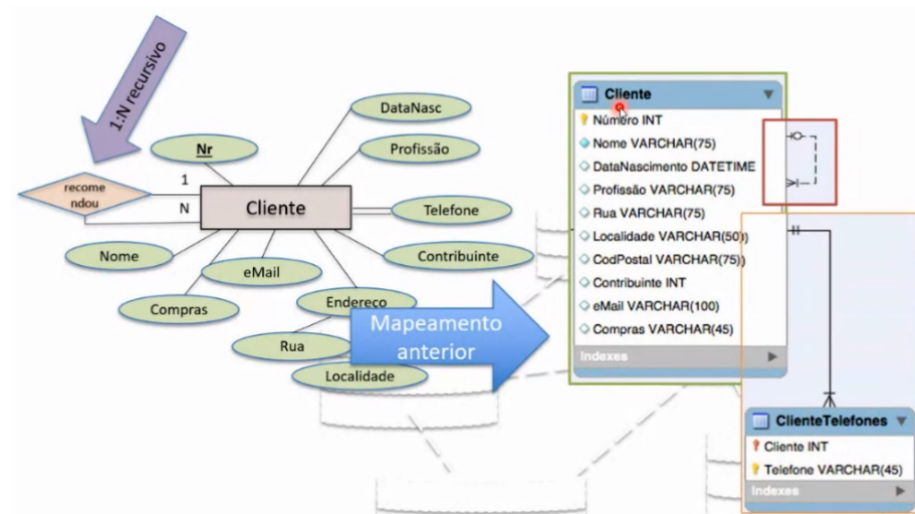


Figure 11: Relacionamentos Recursivos

4.4 Validação do Modelo

A normalização é utilizada para melhorar o modelo de dados já que a aplicação das suas regras permite evitar a duplicação de dados - o desenho lógico pode não ser o desenho final.

A normalização assegura que o modelo resultante é um modelo mais próximo da organização que representa, mais consistente, com um baixo grau de redundância e grande estabilidade.

A elaboração de um esquema normalizado pode ser justificado através das seguintes razões:

- Um desenho normalizado organiza os dados de acordo com as suas dependências funcionais;
- Um desenho normalizado é robusto e está livre de anomalias provocadas por modificações de dados;
- Dada a atual situação dos computadores, por vezes é razoável implementar um desenho que ganha em termos de facilidade de utilização com um custo de processamento adicional;
- A normalização força-nos a entender cada um dos atributos das relações representadas na base de dados;
- A normalização produz um desenho da base de dados flexível, que pode facilmente ser estendido.

4.5 Análise de Interrogações

A análise de interrogações é um passo bastante importante, uma vez que se destina a verificar se as principais (todas) ações de exploração da base de dados, realizadas por queries do sistema ou das aplicações cliente - podem ou não ser satisfeitas.

4.5.1 Exemplos de interrogações

Para o caso de estudo da “Mercearia da D. Acácia” podemos assumir a necessidade de interrogações como as seguintes:

- Quais são os meus clientes?
- Quais foram as vendas realizadas com os clientes ‘João Maria Castro’ e ‘Joana Manuel Pires’?
- Quanto é foi vendido (em valor) durante o último mês?
- etc..

4.6 Análise de Transações

O objectivo deste passo é assegurar que o modelo desenvolvido suporta as transações requeridas pela vista de utilização, definidas com base na especificação dos requisitos dos utilizadores.

4.7 Tipos de Restrições de Integridade

- **Exigência de Dados** - alguns atributos devem conter sempre um valor válido, isto é não devem aceitar valores nulos;
- **Integridade de entidades** - a chave primária de uma relação não pode admitir valores nulos; este tipo de restrição deve ser considerado no momento em que se faz a definição das chaves primárias para cada tipo de entidade;
- **Restrições de domínios de atributos** - qualquer atributo tem um domínio, ou seja um conjunto de valores dos quais o atributo poderá assumir um; estas restrições devem ser identificados no momento da escolha dos domínios;
- **Integridade referencial** - Uma chave estrangeira liga cada ocorrência na relação filho com a respectiva ocorrência na relação pai que contém o valor da chave candidata correspondente. A integridade referencial significa que se uma chave estrangeira contém um valor, esse valor deve existir na relação pai para que seja aceite na chave estrangeira. Os valores nulos devem ser analisados com cuidado, sendo sua utilização dependente do tipo de participação –total ou parcial–da chave estrangeira no relacionamento;
- **Restrições da organização** - Estas restrições, também conhecidas por regras de negócio, são normalmente regras da própria organização que são aplicadas para regulamentar as suas próprias atividades.

4.8 Validar o Modelo de Dados Global

O objectivo desta etapa é a validação do modelo global definido através da sua verificação com base em técnicas de normalização e analisar sobre ele o efeito das transações requeridas.

4.9 Analisar a Evolução da Base de Dados

É importante que o modelo de dados lógico global alcançado possa ser **fácilmente expandido**. Assim, nesta fase espera-se determinar quando é que o no futuro poderão ocorrer modificações e saber estabelecer o momento quando é que o modelo de dados poderá acolher tais modificações com mínimo de impacto possível.

4.10 Versão Final do Modelo Lógico

Nesta fase temos o modelo de dados lógico global completo, por isso podemos desenhar o diagrama lógico final, que representará o modelo de dados lógico global da parte da organização que se pretendeu modelar.