

>>> Grupo de Estudos

>>> Aula 2 - Modelagem de banco de dados

Name: Jose Carlos[†], Gustavo Casarim

Date: 8 de Novembro de 2018

[†]josecarlosdemoraesfilho@gmail.com

```
>>> Conteúdo
```

```
1. Modelos de Dados
```

```
2. Princípios da Modelagem
```

```
3. Normalização - Formas Normais em um Modelo
```


>>> Introdução

- * A modelagem é importante por: Representar o ambiente observado; documentar e normalizar; fornecer os processos de validação; e observar os processos de relacionamentos entre entidades.
- * Existem três tipos de Modelos: Conceitual; Lógico; e Físico.

>>> Modelo Conceitual

- * O Modelo Conceitual representa as regras de negócio, sem limitações tecnológicas ou de implementação, e por isso são adequadas no trato do cliente durante a fase de levantamento de requisitos. Veja um exemplo na Figura 1:

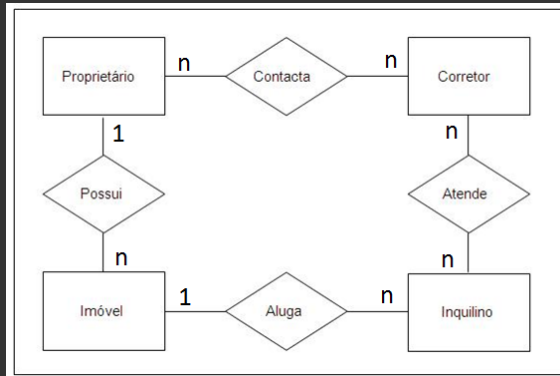


Figura 1: Modelo Conceitual - Exemplo de uma imobiliária

>>> Modelo Conceitual

- * Nesse modelo temos: Visão Geral do negócio; Facilitação do entendimento entre usuários e desenvolvedores; Possui somente as entidades e atributos principais; Possui a representação de relacionamentos.

>>> Modelo Lógico

- * O Modelo Lógico considera limitações impostas por algum tipo de tecnologia de banco de dados (banco de dados hierárquico, relacional, etc), o que inclui domínio de dados (varchar, integer, etc). Veja um exemplo na Figura 2:

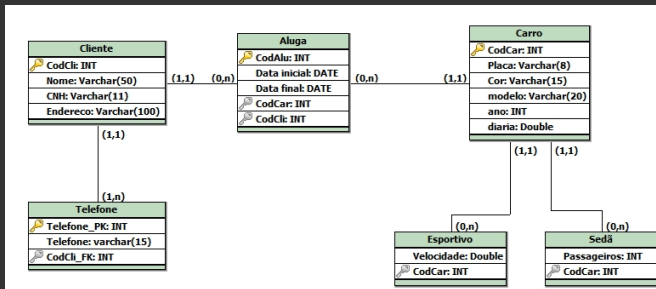


Figura 2: Modelo Lógico - Exemplo de Locadora de Veículos

>>> Modelo Lógico

- * Suas principais características são: Deriva do modelo conceitual e via a representação do negócio; Possui entidades associativas em lugar de relacionamentos n:m; Define as chaves primárias das entidades; Normalização até a 3a. forma normal; Adequação ao padrão de nomenclatura; Entidades e atributos documentados.

>>> Modelo Físico

- * Leva em consideração limites impostos pelo SGBD (Sistema Gerenciador de Banco de dados) e pelos requisitos não funcionais dos programas que acessam os dados. Veja um exemplo na Figura 3:

```
CREATE TABLE `turma` (  
  `idturma` INTEGER(4) NOT NULL AUTO_INCREMENT,  
  `capacidade` INTEGER(2) NOT NULL,  
  `idProfessor` INTEGER(4) NOT NULL,  
  PRIMARY KEY (`idturma`),  
  FOREIGN KEY(`idProfessor`) REFERENCES professor(idProfessor),  
  UNIQUE KEY `idturma` (`idturma`)  
)  
  
CREATE TABLE `professor` (  
  `idProfessor` INTEGER(4) NOT NULL AUTO_INCREMENT,  
  `telefone` INTEGER(10) NOT NULL,  
  `nome` CHAR(80) COLLATE NOT NULL DEFAULT '',  
  PRIMARY KEY (`idProfessor`),  
  FOREIGN KEY(`idTurma`) REFERENCES turma(idturma),  
  UNIQUE KEY `idProfessor` (`idProfessor`)  
)
```

Figura 3: Modelo Físico - Exemplo Turma/Professores

>>> Modelo Físico

- * Suas principais características são: Elaborado a partir do modelo lógico; Pode variar segundo o SGBD; Pode ter tabelas físicas; Pode ter colunas físicas (replicação).

>>> Entidades

- * Uma Entidade pode ser definida como qualquer coisa do mundo real , abstrata ou concreta , na qual se deseja guardar informações. (Tabela , File, etc..). Exemplos de entidades : Cliente , Produto , Contrato , Vendas , etc.
- * Uma Entidade pode ser: Forte; Fraca; ou Associativa.
- * Entidade Forte: são aquelas cuja existência independe de outras entidades, ou seja, por si só elas já possuem total sentido de existir. Em um sistema de vendas, a entidade produto, por exemplo, independe de quaisquer outras para existir.

>>> Entidades

- * Entidade Fraca: ao contrário das entidades fortes, as fracas são aquelas que dependem de outras entidades para existirem, pois individualmente elas não fazem sentido. Mantendo o mesmo exemplo, a entidade venda depende da entidade produto, pois uma venda sem itens não tem sentido.
- * Entidade Associativa: Esse tipo de entidade surge quando há a necessidade de associar uma entidade a um relacionamento existente. Surge em relacionamentos N:M.

>>> Atributos

- * Atributos definem as características de uma entidade;
- * Em termos de função podem ser classificados como:
Descritivos; Nominativos; ou Referenciais;
- * Em termos estruturais podem ser classificados como:
Simples; ou Compostos;
- * Atributos podem ser obrigatórios, ou opcionais.

>>> Atributos

- * Um atributo identificador imprescindível para uma entidade é a Chave Primária;
- * Esse atributo é capaz de identificar exclusivamente cada ocorrência de uma entidade. Também conhecido como chave Primária ou Primary Key (PK). Ex: Código do Cliente , Código do Produto , etc;
- * Chave Candidata: Atributo ou grupamento de atributos que têm a propriedade de identificar unicamente uma ocorrência da entidade . Pode vir a ser uma chave Primária. A chave candidata que não é chave primária também chama-se chave Alternativa. Um exemplo seria o CPF;

>>> Relacionamentos

- * Relacionamentos indicam a relação entre as entidades, de modo a explicitar a quantidade de objetos envolvidos em cada relação. Daí surge o conceito de cardinalidade;
- * A Cardinalidade indica quantas ocorrências de uma Entidade participam no mínimo e no máximo do relacionamento; Existem 3 tipos de cardinalidades possíveis: Relacionamento 1..1 (um para um); Relacionamento 1..n ou 1..* (um para muitos); e Relacionamento n..n ou *..* (muitos para muitos).

>>> ACID

- * As formas normais surgem da necessidade de atender à premissa de um banco de dados que possua transações ACID;
- * ACID é um acrônimo para: Atomicidade, Consistência, Isolamento e Durabilidade.
- * Atomicidade: Em uma transação envolvendo duas ou mais partes de informações discretas, ou a transação será executada totalmente ou não será executada, garantindo assim que as transações sejam atômicas.
- * Consistência: A transação cria um novo estado válido dos dados ou em caso de falha retorna todos os dados ao seu estado anterior à transação.

>>> ACID

- * Isolamento: Uma transação em andamento mas ainda não validada deve permanecer isolada de qualquer outra operação, ou seja, garantimos que a transação não será interferida por nenhuma outra transação concorrente.
- * Durabilidade: Dados validados são registrados pelo sistema de tal forma que mesmo no caso de uma falha e/ou reinício do sistema, os dados estão disponíveis em seu estado correto.

>>> ACID - Exemplo de Anomalias

* Na Figura 4, vemos um tabela não normatizada:



	Codigo	Nome	Cargo	Setor	QuantidadeFuncionarios
	1	Miriam	Gerente	Vendas	23
	2	Jefferson	Programador	Suporte	20
	3	Jessica	Analista	Compras	15
	4	Janice	Programadora	Suporte	16
	5	Mario	Gerente	Design	9

Figura 4: ACID - Tabela não normatizada

>>> ACID - Exemplo de Anomalias

- * Anomalia de Exclusão - O que acontece se você excluir o funcionário de código igual a 3 ? O Setor vai ser excluído junto;
- * Anomalia de Alteração - O nome do Setor Suporte mudou para Apoio . Você terá que alterar o nome em todos os registros da tabela;
- * Anomalia de Inclusão - Foi contratado um novo funcionário para o Setor Suporte. Você vai ter que incluir um funcionário ao campo - QuantidadeFuncionarios - em todas as ocorrências com setor de nome SUPORTE.

>>> Primeira Forma Normal - 1FN

- * Uma relação está na 1FN se somente todos os domínios básicos contiverem somente valores atômicos (não contiver grupos repetitivos). Para atingir esta forma normal devemos eliminar os grupos de repetição;
- * Procedimentos: a) Identificar a chave primária da entidade; b) Identificar o grupo repetitivo e excluí-lo da entidade; c) Criar uma nova entidade com a chave primária da entidade anterior e o grupo repetitivo;

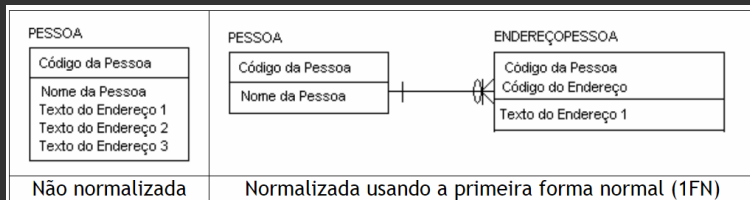


Figura 5: 1FN - Primeira Forma Normal

>>> Segunda Forma Normal - 2FN

- * Uma relação R está na 2FN se e somente se ela estiver na primeira e todos os atributos não chave forem totalmente dependentes da chave primária (dependente de toda a chave e não apenas de parte dela);
- * Procedimentos: a) Identificar os atributos que não são funcionalmente dependentes de toda a chave primária; b) Remover da entidade todos esses atributos identificados e criar uma nova entidade com eles;



Figura 6: 2FN - Segunda Forma Normal

>>> Terceira Forma Normal - 3FN

- * Uma relação R está na 3FN se somente estiver na 2FN e todos os atributos não chave forem dependentes não transitivos da chave primária (cada atributo for funcionalmente dependente apenas dos atributos componentes da chave primária ou se todos os seus atributos não chave forem independentes entre si);
- * Procedimentos: a) Identificar todos os atributos que são funcionalmente dependentes de outros atributos não chave; b) Removê-los e criar uma nova entidade com os mesmos;
- * Considerando a Figura 6, criaríamos uma entidade: Cliente(Código do Cliente, Nome do cliente, Endereço do cliente);

>>> Referências

[Mac18] [Dev18] [Bar16]



Pedro Barros, *O que é acid?*, 2016, Last accessed at November 08 2018.



DevMedia, *Definindo mer (modelo entidade relacionamento) e der (diagrama entidade-relacionamento)*, 2018, Last accessed at November 08 2018.



Macoratti, *Conceitos básicos de modelagem de dados*, 2018, Last accessed at November 08 2018.