


3 - Introdução

- A modelagem conceitual é muito abstrata, tanto que profissionais mais experientes usam somente em casos de modelagem mais complexos.
- Já o modelo lógico é mais próximo de como o banco de dados será implementado realmente, já possuindo conceitos práticos, como tabelas, chaves primárias e estrangeiras.
- Após o modelo lógico estar pronto, a sua transformação para o modelo físico é um processo extremamente fácil.

3.1 - Tabelas


- As tabelas são a base do banco de dados relacional, pois os dados são armazenados nelas. São formadas por linhas e colunas, onde cada coluna possui um nome e um tipo de dados específico. As figuras abaixo mostram como uma tabela é representada no modelo lógico e como os dados são organizados:

Produtos		
id_produto	descricao	preco
1	Pen Drive	30
2	Mouse	15
3	Teclado	20

Produto
 id_produto: Número(4)
descricao: Texto(50)
preco: Número(10)

3.2 - Chave Primária

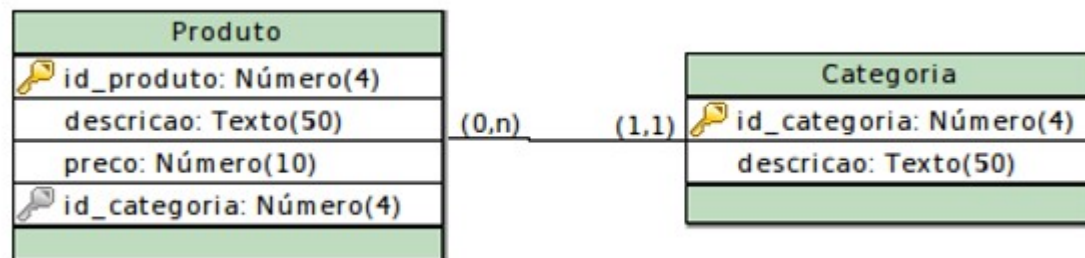
- Uma chave primária é uma coluna ou uma combinação de colunas responsável pela identificação dos registros de uma tabela.
- A Chave Primária deve ser formada por um ou mais campos cujos valores nunca se repetem, desta forma será possível identificar cada um dos registros pelo valor de sua chave. Toda tabela deve possuir uma chave primária.
- No caso da de produtos, o campo id_produto seria a chave primária, sendo que o próprio SGBD não aceitará valores repetidos neste campo:

Produto	
	id_produto: Número(4)
	descricao: Texto(50)
	preco: Número(10)

Produtos		
id_produto	descricao	preco
1	Pen Drive	30
2	Mouse	15
3	Teclado	20

3.3 - Chave Estrangeira

- A chave estrangeira é o mecanismo que permite a implementação de relacionamentos em um banco de dados relacional.
- É possível definir chave estrangeira como um campo que é chave primária em uma tabela, e que também existe em uma outra tabela que está relacionada com a primeira (nesse caso esse campo da segunda tabela é chamado de chave estrangeira).
- No exemplo abaixo, o campo id_categoria da tabela de produto é a chave estrangeira, visto que não tabela de Categoria, é chave primária:



3.3 - Chave Estrangeira

- O relacionamento entre as tabelas feitos a partir da chave estrangeira é importante para garantir a integridade dos dados.
- A tabela abaixo mostra os problemas que podem surgir quando o relacionamento não é feito, pois o usuário pode cadastrar diferentes nomes para a mesma cidade, ou ainda pode cometer erros de digitação:

Cliente		
id_cliente	nome	cidade
1	Robinho	Guarapuava
2	Dudu	Curitiba
3	Gabriel Jesus	GP
4	Arouca	Guara
5	Edmundo	<u>Curitibo</u>

3.3 - Chave Estrangeira

- Quando as tabelas estão relacionadas, o banco de dados garante que não existam inconsistências, pois será criada uma tabela de cidade que será relacionada com cliente. Desta forma, somente poderão ser usadas as cidades cadastradas, como mostra o exemplo abaixo:

Cliente				Cidade	
id_cliente	nome	id_cidade		id_cidade	nome
1	Robinho	1		1	Guarapuava
2	Dudu	2		2	Curitiba
3	Gabriel Jesus	1		3	Ponta Grossa
4	Arouca	1		4	Maringá
5	Edmundo	3			

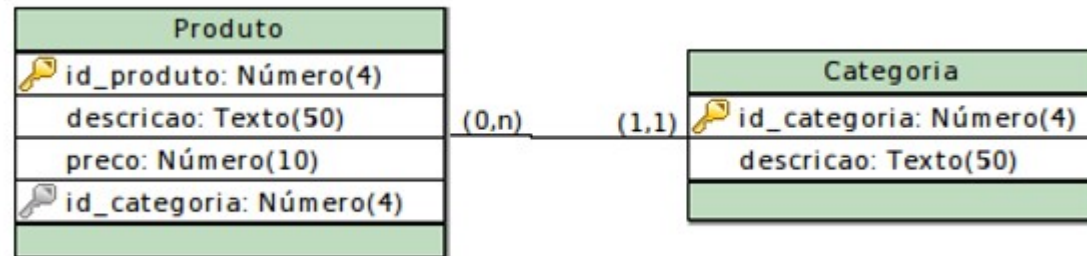
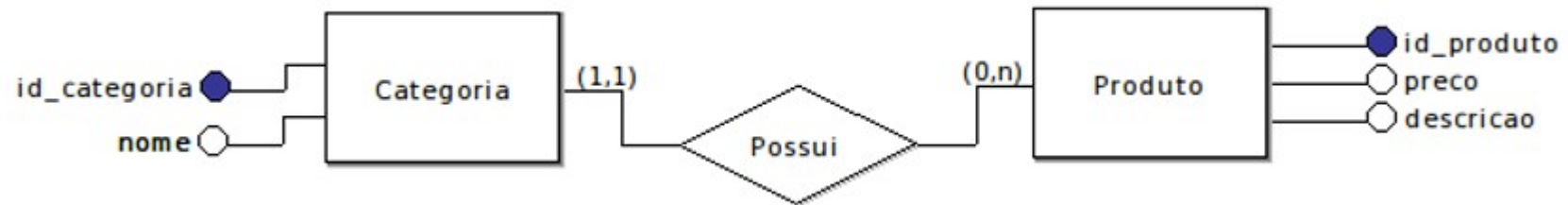
3.4 - Integridade Referencial

- Um dos objetivos primordiais de um SGBD é a integridade de dados. Desta forma, os SGBD's oferecem os mecanismos de restrições de integridade, responsáveis por garantir que os dados inseridos no banco são válidos. Existem quatro tipos de restrições de integridade:

1. **Integridade de domínio:** As colunas pode receber dados somente dentro do tipo pré-definido (número inteiro, número real, alfanumérico, data, etc.).
2. **Integridade de vazio:** Essa restrição de integridade determina se uma coluna pode ou não ter valores nulos (vazio).
3. **Integridade de chave:** Garante que os valores da chave primária sejam únicos.
4. **Integridade referencial:** É a restrição que garante que os valores de uma coluna definida como chave estrangeira tenha um correspondente na chave primária da tabela referenciada.

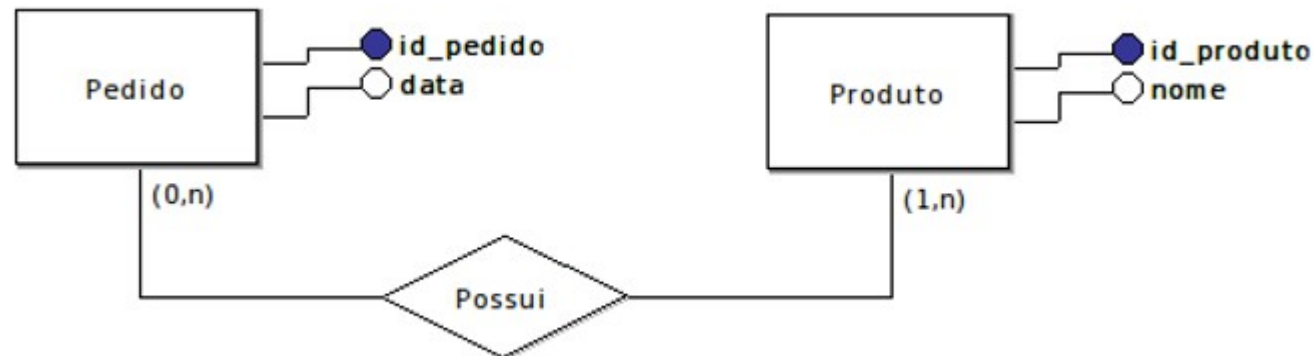
3.5 - Transformação do Conceitual para Lógico

- As figuras abaixo mostram a correspondência entre os modelos conceitual e lógico:



3.6 - Tratando a Cardinalidade N:N

- A cardinalidade N:N somente pode ser feita no modelo conceitual, pois não é possível implementá-la no banco de dados, logo não é possível representar o N:N nos modelos lógico e físico.
- Para resolver este problema, a cardinalidade N:N deve ser quebrada em duas 1:N. Para isso, deverá ser criada uma tabela intermediária, como mostra o exemplo abaixo:



3.6 - Tratando a Cardinalidade N:N

- Nesta transformação para o modelo lógico, uma tabela intermediária, chamada item_pedido será criada para quebrar o N:N em dois relacionamentos 1:N. A tabela intermediária deverá ter como chave estrangeira as chaves primárias das tabelas envolvidas.

