

ADIÇÃO DE COLUNAS 1:N

FORNECEDOR(**id_fornecedor**, nome, contato)

PRODUTOS(**id_produto**, estoque, nome, valor, **id_fornecedor**)

id_fornecedor referência FORNECEDOR

ADIÇÃO DE COLUNAS 1:N

CATEGORIA(**id_categoria**, nome)

PRODUTOS(**id_produto**, estoque, nome, valor, **id_fornecedor**,
id_categoria)

id_categoria referência CATEGORIA

ADIÇÃO DE COLUNAS 1:N

SUPERMERCADO (**id_supermercado**, nome, contato, endereco)

PEDIDOS (**id_pedido**, quantidade, **id_supermercado**)

id_supermercado referência SUPERMERCADO

TABELA PRÓPRIA N:N

PEDIDO (**id_pedido**, quantidade)

PRODUTOS (**id_produto**, estoque, nome, valor)

PEDIDO_PRODUTO (**id_produto**, **id_pedido**)

id_produto referência PRODUTO

id_pedido referência PEDIDO

Estabelecemos uma relação 1:N entre Fornecedor e Produtos. Para isso, foram criadas as tabelas FORNECEDOR e PRODUTOS, onde a tabela PRODUTOS recebeu a chave estrangeira id_fornecedor, permitindo a associação de vários Produtos a um único Fornecedor. Do mesmo modo, na relação 1:N entre Categoria e Produtos, foram criadas as tabelas CATEGORIA e PRODUTOS, com a inclusão da chave estrangeira id_categoria na tabela PRODUTOS, indicando que um Produto pode pertencer a uma Categoria específica. Além disso, implementou-se a relação 1:N entre Supermercado e Pedidos, sendo necessário criar as tabelas SUPERMERCADO e PEDIDOS, com a adição da chave estrangeira id_supermercado na tabela PEDIDOS, deixando claro que um Supermercado pode ter vários Pedidos associados. Por fim, criou-se uma tabela de relacionamento N:N entre Pedidos e Produtos, que é representada pelas tabelas PEDIDO, PRODUTOS e PEDIDO_PRODUTO, onde houve a implementação da última tabela para resolver o problema N:N, em que ela recebeu id_pedido e id_produto. Com essa implementação, foi possível que um Pedido possa conter vários Produtos e vice-versa.