

Loja (idLoja, nome, endereco, telefone)

Item (idItem, nome, dataEntrada, categoria, dimensoes, material, precoVenda, idLoja -> Loja)

Estado (idEstado, tipo, dataAtribuicaoEstado, idItem -> Item)

TermoEmprestimo (idEmprestimo, valorEmprestimo, valorDevolverComJuros, dataInicioEmprestimo, dataLimitePagamento, idItem -> Item)

Transacao (idItem -> Item, nifCliente -> Cliente, tipo, valor, dataTransacao, metodoPagamento)

Avaliacao (idItem -> Item, nifFuncionario -> Funcionario, dataAvaliacao, notasAvaliacao, valorAvaliado, estadoConservacao)

Pessoa (nif, nome, telefone, email)

Funcionario (nifFuncionario -> Pessoa, salario, idLoja -> Loja)

Cliente (nifCliente -> Pessoa)

—

IA utilizada: Claude.ai

Output da IA: *****

R1: FUNCIONARIO (_NIF_, nome, numero_telemovel, email, salario, id_loja->LOJA)

R2: CLIENTE (_NIF_, nome, numero_telemovel, email)

R3: LOJA (_id_, nome, endereco, telefone)

R4: ITEM (_id_, nome, categoria, dimensoes, material, preco_venda, data_entrada, id_loja->LOJA)

R5: ESTADO (_id_, tipo, data_atribuicao_estado, id_item->ITEM)

R6: AVALIACAO (_id_, data_avaliacao, notas_avaliacao, valor_avaliado, estado_conservacao, id_item->ITEM, nif_funcionario->FUNCIONARIO)

R7: TERMO_EMPRESTIMO (_id_, valor_emprestimo, valor_devolver_com_juros, data_inicio_emprestimo, data_limite_pagamento, id_item->ITEM)

R8: TRANSACAO (_id_, tipo, valor, data_transacao, metodo_pagamento, nif_cliente->CLIENTE, id_item->ITEM)

Ela propôs, basicamente, o mesmo esquema relacional (embora se tenha enganado ligeiramente primeiro e tenha sido necessário corrigi-la). As únicas diferenças são relativas a alguns ids, entretanto, em algumas das classes, as chaves estrangeiras servem

perfeitamente de chave primária (composta). Como tal, mantém-se o esquema inicialmente proposto.

— — —

Loja:

idLoja -> nome, endereco, telefone

Item:

idItem -> nome, dataEntrada, categoria, dimensoes, material, precoVenda, idLoja

Estado:

idEstado -> tipo, dataAtribuicaoEstado, idItem

TermoEmprestimo:

idEmprestimo -> valorEmprestimo, valorDevolverComJuros, dataInicioEmprestimo, dataLimitePagamento, idItem

Transacao:

idItem, nifCliente -> tipo, valor, dataTransacao, metodoPagamento

Avaliacao:

idItem, nifFuncionario -> dataAvaliacao, notasAvaliacao, valorAvaliado, estadoConservacao

Funcionario:

nifFuncionario -> nome, telefone, email, salario, idLoja

Cliente:

nifCliente -> nome, telefone, email

Para uma dependência funcional não trivial estar na Forma Normal de Boyce-Codd, o seu lado esquerdo deve ser uma superchave. Visto que a chave de uma relação é também superchave, pode afirmar-se que todas as relações apresentadas estão na Forma Normal de Boyce-Codd.

Uma dependência funcional não trivial está na Terceira Forma Normal quando o seu lado esquerdo é uma superchave ou o seu lado direito consiste somente em atributos primos. Uma vez que todas as dependências funcionais identificadas cumprem ambas as condições, é possível afirmar que todas as relações apresentadas estão na Terceira Forma Normal.

—

IA utilizada: Claude.ai

Output da IA:

A mesmíssima coisa.

Loja:

- idLoja -> {nome, endereco, telefone}

Item:

- idItem -> {nome, dataEntrada, categoria, dimensoes, material, precoVenda, idLoja}

Estado:

- idEstado -> {tipo, dataAtribuicaoEstado, idItem}

TermoEmprestimo:

- idEmprestimo -> {valorEmprestimo, valorDevolverComJuros, dataInicioEmprestimo, dataLimitePagamento, idItem}

Transacao:

- {idItem, nifCliente} -> {tipo, valor, dataTransacao, metodoPagamento}

Avaliacao:

- {idItem, nifFuncionario} -> {dataAvaliacao, notasAvaliacao, valorAvaliado, estadoConservacao}

Funcionario:

- nifFuncionario -> {nome, telefone, email, salario, idLoja}

Cliente:

- nifCliente -> {nome, telefone, email}
-

[...]

Conclusão: Todas as relações estão na Forma Normal de Boyce-Codd (BCNF) e, consequentemente, também estão na 3NF, pois:

1. Todas as DFs têm superchaves como determinantes
 2. Não existem dependências transitivas
 3. Não existem dependências parciais
-