

Bases de Dados

Stand de Automóveis

2LEIC06 - Grupo 603

(7 de novembro de 2023)

David Carvalho up202208654@fe.up.pt

Leonardo Magalhães up202208726@fe.up.pt

Tiago Pinto up202206280@fe.up.pt

Descrição

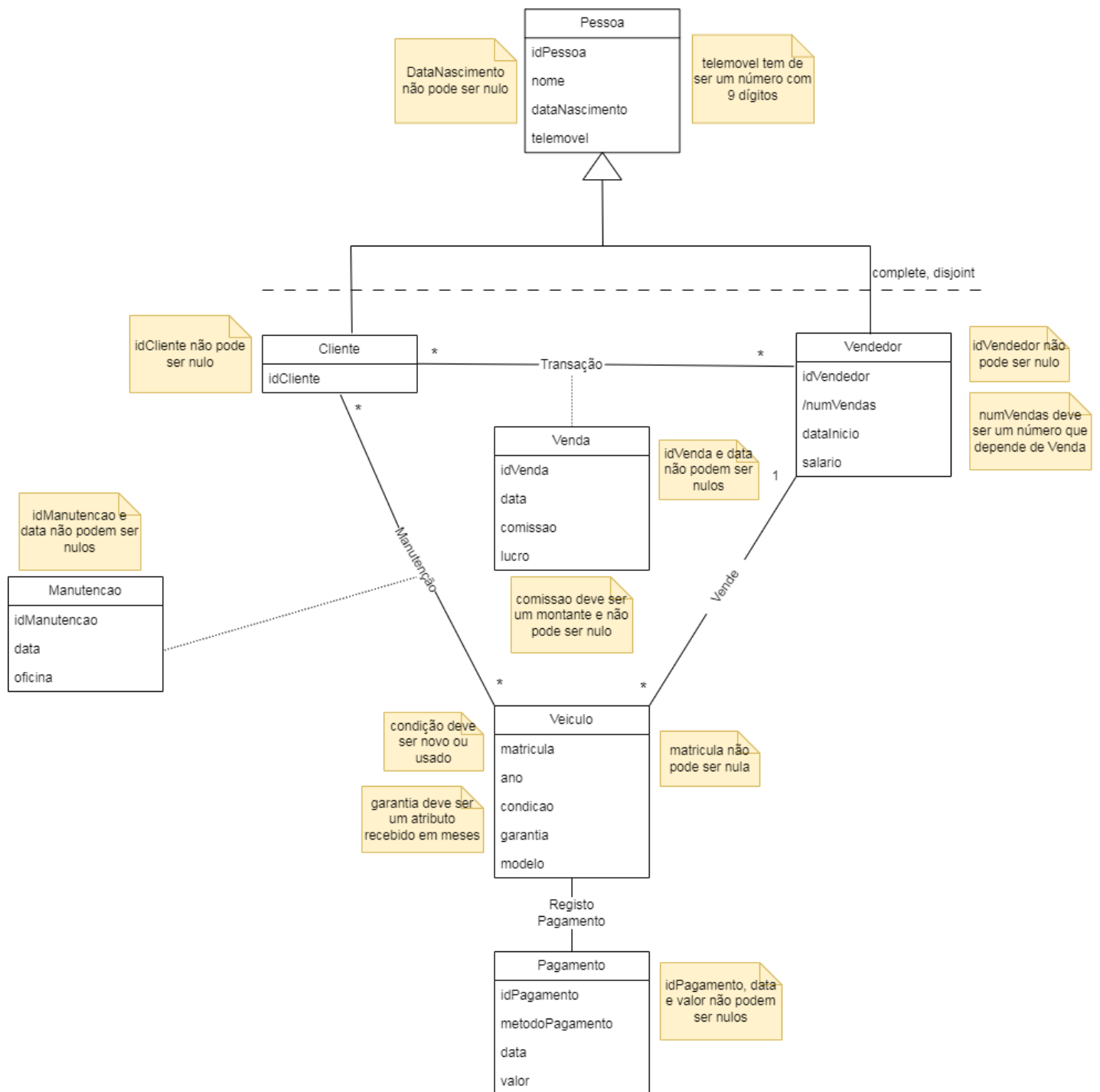
Um stand de automóveis pretende informatizar o seu serviço de **vendas**. É necessário armazenar informações de cada **pessoa**, o nome, a data de nascimento e o telemóvel, que tanto pode ser um **vendedor** do stand, onde se pretende guardar o número de vendas, o salário e data que começou a trabalhar no stand, como também um **cliente**. Este possui um serviço de **manutenção**, que tem uma data de início e oficina associada. Interessa também saber a matrícula, o ano, a garantia, condição e modelo de cada **veículo**. Os **vendedores** são responsáveis pela **venda** do **veículo** ao **cliente**, sendo necessário guardar a data da **venda**, lucro e comissão correspondente. Por último, é relevante guardar informações sobre o **pagamento**, como o seu método, data e valor.

A informatização do serviço de vendas permite realizar diversas consultas importantes, tais como, por exemplo, calcular a média de vendas, determinar o lucro de cada venda, calcular a média de lucro por vendedor, identificar quais veículos foram vendidos, exibindo informações como data da venda, preço de venda e lucro e obter o valor do pagamento e o método utilizado em cada venda.

Também pode incidir sobre outras áreas, como verificar quantos carros de um modelo específico estão atualmente em stock, registar a oficina onde cada veículo realizou manutenção, calcular o gasto mensal com salários, determinar quantos veículos um cliente específico comprou ao longo do tempo, verificar a garantia de um veículo e quantas vezes ele foi submetido a manutenção.

Essas consultas contribuem significativamente para o funcionamento eficaz do stand de automóveis, ajudando a manter um controlo financeiro, auxiliando a compreensão do histórico de manutenção de cada veículo e permitindo uma gestão eficiente das vendas, dos custos e da satisfação do cliente.

Diagrama de Classes em UML - versão atualizada



Modelo Relacional - versão inicial

Cliente(idCliente, nome, dataNascimento, telemovel)

Vendedor(idVendedor, nome, numVendas, dataNascimento, dataInicio, salario, telemovel)

Venda(idCliente->Cliente, idVendedor->Vendedor, data, comissao, lucro)

Veiculo(matricula, ano, condicao, garantia, modelo, idPagamento->Pagamento)

VeiculoVendedor(matricula->Veiculo, idVendedor->Vendedor)

Manutencao(matricula->Veiculo, idCliente->Cliente, idManutencao, data, oficina)

Pagamento(idPagamento, metodoPagamento, data, valor, matricula->Veiculo)

Modelo Relacional - integração de AI

Para nos auxiliar e otimizar o modelo relacional, usamos o ChatGPT, que é um software que utiliza inteligência artificial para processar e gerar texto com base em padrões e informações pré-existentes.

Para tal, fizemos a seguinte instrução:

Considere o seguinte modelo relacional:

Cliente(idCliente, nome, dataNascimento, telemovel)

Vendedor(idVendedor, nome, numVendas, dataNascimento, dataInicio, salario, telemovel)

Venda(idCliente->Cliente, idVendedor->Vendedor, data, comissao, lucro)

Veiculo(matricula, ano, condicao, garantia, modelo, idPagamento->Pagamento)

VeiculoVendedor(matricula->Veiculo, idVendedor->Vendedor)

Manutencao(matricula->Veiculo, idCliente->Cliente, idManutencao, data, oficina)

Pagamento(idPagamento, metodoPagamento, data, valor, matricula->Veiculo)

Melhore este modelo, corrigindo alguns possíveis erros

O ChatGPT deu a seguinte resposta:

Aqui está uma versão revisada do modelo:

```
Pessoa(idPessoa, nome, dataNascimento, telemovel)
Cliente(idCliente, idPessoa->Pessoa)
Vendedor(idVendedor, numVendas, dataInicio, salario, idPessoa->Pessoa)
Venda(idVenda, idCliente->Cliente, idVendedor->Vendedor, data, comissao, lucro)
Veiculo(matricula, ano, condicao, garantia, modelo)
VeiculoVendedor(matricula->Veiculo, idVendedor->Vendedor)
Manutencao(idManutencao, matricula->Veiculo, idCliente->Cliente, data, oficina)
```

Neste modelo, as informações pessoais são mantidas em uma tabela Pessoa, que é compartilhada por Clientes e Vendedores. Além disso, as chaves primárias foram adicionadas, garantindo a unicidade dos registros. A tabela Pagamento foi removida com base na suposição de que os pagamentos estão diretamente relacionados a vendas e podem ser rastreados na tabela Venda.

Com base nesta resposta, decidimos fazer algumas alterações no nosso modelo inicial. Achemos que, realmente, faz sentido a integração da classe Pessoa, devido aos vários atributos comuns entre as classes Cliente e Vendedor. Contudo, decidimos não retirar a tabela de pagamentos, dado que esta pode servir como histórico de todas as transações realizadas.

Em relação ao output da AI, não inclui o texto sublinhado, pelo que não identifica as primary/foreign keys. Contudo, foi bastante útil, dado que conseguiu encontrar e explicar, de forma eficaz, algumas redundâncias presentes no modelo inicial.

Por fim, alteramos o diagrama UML e o modelo relacional. O novo modelo relacional é o seguinte:

```
Pessoa(idPessoa, nome, dataNascimento, telemovel)
Cliente(idCliente, idPessoa->Pessoa)
Vendedor(idVendedor, numVendas, dataInicio, salario, idPessoa->Pessoa)
Venda(idCliente->Cliente, idVendedor->Vendedor, data, comissao, lucro)
Veiculo(matricula, ano, condicao, garantia, modelo, idPagamento->Pagamento)
VeiculoVendedor(matricula->Veiculo, idVendedor->Vendedor)
Manutencao(matricula->Veiculo, idCliente->Cliente, idManutencao, data, oficina)
Pagamento(idPagamento, metodoPagamento, data, valor, matricula->Veiculo)
```

Dependências Funcionais e Análise da Forma Normal

Vamos analisar estas relações e identificar as dependências funcionais, violações da Forma Normal de Boyce-Codd (FNBC) e violações da Terceira Forma Normal (3FN) em cada uma das tabelas:

1. Pessoa(idPessoa, nome, dataNascimento, telemovel)

- Dependências Funcionais: Não existem dependências funcionais além das já expressas pelas colunas.
- Violação da FNBC: Não há violações da FNBC, uma vez que cada coluna está funcionalmente dependente da chave primária (idPessoa).
- Violação da 3FN: Não há violações da 3FN, uma vez que todas as colunas estão diretamente relacionadas à chave primária (idPessoa).

2. Cliente(idCliente, idPessoa->Pessoa)

- Dependências Funcionais: idCliente -> idPessoa
- Violação da FNBC: Não há violações da FNBC, pois a chave primária (idCliente) determina diretamente a coluna idPessoa, que, por sua vez, determina a coluna nome, dataNascimento e telemovel.
- Violação da 3FN: Não há violações da 3FN, pois a dependência entre idCliente e idPessoa não gera problemas de transitividade.

3. Vendedor(idVendedor, numVendas, dataInicio, salario, idPessoa->Pessoa)

- Dependências Funcionais: idVendedor -> idPessoa
- Violação da FNBC: Não há violações da FNBC, pois a chave primária (idVendedor) determina diretamente a coluna idPessoa, que, por sua vez, determina as colunas nome, dataNascimento e telemovel.
- Violação da 3FN: Não há violações da 3FN, pois a dependência entre idVendedor e idPessoa não gera problemas de transitividade.

4. Venda(idCliente->Cliente, idVendedor->Vendedor, data, comissao, lucro)

- Dependências Funcionais:

- idCliente -> idVendedor, data, comissao, lucro

- idVendedor -> idCliente, data, comissao, lucro

- Violação da FNBC: Não há violações da FNBC, pois todas as colunas são funcionalmente dependentes da chave primária (idCliente, idVendedor).

- Violação da 3FN: Não há violações da 3FN, pois as dependências funcionais não geram problemas de transitividade.

5. Veiculo(matricula, ano, condicao, garantia, modelo, idPagamento->Pagamento)

- Dependências Funcionais:

- matricula -> ano, condicao, garantia, modelo, idPagamento

- idPagamento -> metodoPagamento, data, valor

- Violação da FNBC: Não há violações da FNBC, pois todas as colunas são funcionalmente dependentes da chave primária (matricula).

- Violação da 3FN: Não há violações da 3FN, pois as dependências funcionais não geram problemas de transitividade.

6. VeiculoVendedor(matricula->Veiculo, idVendedor->Vendedor)

- Dependências Funcionais: Não existem dependências funcionais além das já expressas pelas colunas.

- Violação da FNBC: Não há violações da FNBC, uma vez que cada coluna está funcionalmente dependente da chave primária (matricula, idVendedor).

- Violação da 3FN: Não há violações da 3FN, uma vez que todas as colunas estão diretamente relacionadas às chaves primárias (matricula, idVendedor).

7. Manutencao(matricula->Veiculo, idCliente->Cliente, idManutencao, data, oficina)

- Dependências Funcionais: Não existem dependências funcionais além das já expressas pelas colunas.

- Violação da FNBC: Não há violações da FNBC, uma vez que cada coluna está funcionalmente dependente das chaves primárias (matricula, idCliente, idManutencao).

- Violação da 3FN: Não há violações da 3FN, pois as colunas são diretamente relacionadas às chaves primárias (matricula, idCliente, idManutencao).

8. Pagamento(idPagamento, metodoPagamento, data, valor, matricula->Veiculo)

- Dependências Funcionais: Não existem dependências funcionais além das já expressas pelas colunas.

- Violação da FNBC: Não há violações da FNBC, uma vez que cada coluna está funcionalmente dependente da chave primária (idPagamento).

- Violação da 3FN: Não há violações da 3FN, uma vez que todas as colunas estão diretamente relacionadas à chave primária (idPagamento).

Com base na análise acima, todas as relações estão na Forma Normal de Boyce-Codd (FNBC) e na Terceira Forma Normal (3FN). Portanto, não é necessário decompor nenhuma relação.

Dependências Funcionais e Análise da Forma Normal - integração de AI

Apresentamos a seguinte instrução ao ChatGPT:

```
Pessoa(idPessoa, nome, dataNascimento, telemovel)
Cliente(idCliente, idPessoa->Pessoa)
Vendedor(idVendedor, numVendas, dataInicio, salario, idPessoa->Pessoa)
Venda(idCliente->Cliente, idVendedor->Vendedor, data, comissao, lucro)
Veiculo(matricula, ano, condicao, garantia, modelo, idPagamento->Pagamento)
VeiculoVendedor(matricula->Veiculo, idVendedor->Vendedor)
Manutencao(matricula->Veiculo, idCliente->Cliente, idManutencao, data, oficina)
Pagamento(idPagamento, metodoPagamento, data, valor, matricula->Veiculo)
```

Verifique a 3 Forma Normal e a Forma Normal Boyce-Codd.

A ferramenta AI confirmou que a 3ª forma normal está verificada. Contudo, apesar de aparentemente estar correto, não consegue garantir a forma normal de Boyce-Codd devido à necessidade de considerar dependências funcionais adicionais e a complexidade das relações no modelo, o que pode requerer intervenção humana. Desta forma, a utilização de AI foi útil para garantir que a implementação do modelo relacional não viola a forma normal.

Criação de código SQL- Integração de AI

Após a realização do modelo relacional, pedi ao chat gpt para criar código em sql tendo em conta o modelo relacional da nossa base de dados. Posto isto, a resposta dele foi muito interessante e muito útil, visto que nos deu um código muito sucinto, claro e que, na sua maior parte, é perfeitamente utilizável.

No entanto, após a leitura do código e após aplicar um sentido crítico ao mesmo, apercebemo-nos que faltavam alguns detalhes ao código, como por exemplo:

- Restringir os valores permitidos para números de telemóvel.
- Restringir as sequências permitidas para as matrículas.
- Restringir todos os Id 's, para que cada um seja único, garantindo assim a unicidade dos mesmos.