

”

E-fólio B | Folha de resolução para E-fólio

UNIDADE CURRICULAR: Fundamentos de Bases de Dados

CÓDIGO: 21053

DOCENTE: Professor: Paulo Pombinho; Tutores: Águeda Ramos, Fábio Morais

A preencher pelo estudante

NOME: Yrma Marina Vianez Martins

N.º DE ESTUDANTE: 2300212

CURSO: Engenharia Informática

TRABALHO / RESOLUÇÃO:

1.a)

```
SELECT Nome, Categoria  
FROM Produtos  
WHERE Preço > (SELECT AVG(Preço) FROM Produtos);
```

A consulta SQL apresentada utiliza uma subconsulta para calcular a média dos preços de todos os produtos existentes na tabela Produtos. Esta subconsulta,

`(SELECT AVG(Preço) FROM Produtos)`, é incorporada na cláusula `WHERE` da consulta principal, devolvendo o valor que corresponde à média dos preços. De seguida, a condição `Preço >` é utilizada para filtrar os produtos cujo preço individual seja superior à média calculada. A consulta principal seleciona apenas as colunas `Nome` e `Categoria` dos produtos que cumprem essa condição, garantindo que apenas as informações relevantes sejam apresentadas.

1.b)

A tradução da consulta SQL para álgebra relacional pode ser dividida em três etapas principais, que posteriormente são consolidadas numa única expressão final.

- Na primeira etapa usa-se uma operação de agregação(γ), para calcular a média de preços: $\gamma_{AVG(Preço)}(Produtos)$, onde a média da coluna `Preço` é calculada.
- Na segunda etapa, selecionam-se os produtos cujo preço é superior à média calculada. Esta seleção é realizada pela operação:

$\sigma_{Preço > (\gamma_{AVG(Preço)}(Produtos))}$, que filtra as linhas da tabela Produtos, retendo apenas aquelas onde o valor do preço excede a média.

- Na terceira etapa, realiza-se a projeção das colunas relevantes. Nesta operação, representada por $\pi_{Nome, Categoria} \left(\sigma_{Preço > (\gamma_{AVG(Preço)}(Produtos))} (Produtos) \right)$, extraem-se apenas as colunas `Nome` e `Categoria` da tabela resultante da seleção, eliminando as restantes colunas, como `ID_Produto`, `Preço` e `Quantidade`.

2.a)

```
CREATE ROLE Caixa;
```

```
CREATE ROLE Gerente;
```

Estes comandos são utilizados para criar papéis no sistema, permitindo a centralização e organização das permissões.

2.b)

```
GRANT SELECT ON Contas TO Caixa;
```

O comando `GRANT SELECT` concede ao papel "Caixa" a permissão de consulta (leitura) na tabela `Contas`. Ao especificar apenas a permissão `SELECT`, é garantido que o papel não terá permissões de escrita, como inserção, atualização ou eliminação de dados.

2.c)

```
GRANT SELECT, INSERT, UPDATE ON Clientes TO Gerente;
```

O comando `GRANT SELECT, INSERT, UPDATE` atribui ao papel "Gerente" as permissões necessárias para consultar (`SELECT`), inserir (`INSERT`) e atualizar (`UPDATE`) dados na tabela `Clientes`. Ao não incluir a permissão `DELETE`, garantimos que o papel não pode eliminar registros.

2.d)

```
GRANT Caixa TO Gerente WITH GRANT OPTION;
```

O comando atribui ao papel "Gerente" a capacidade de utilizar o papel "Caixa". A cláusula `WITH GRANT OPTION` permite ao "Gerente" conceder o papel "Caixa" a outros utilizadores.

3)

A tabela inicial não está em nenhuma Forma Normal, porque contém colunas multi-valoradas ou não atômicas, como a coluna Produtos, que inclui múltiplos produtos com nome, quantidade, categoria e preço_unitário.

Primeira Forma Normal: para atingir a 1FN, todos os atributos devem conter valores atômicos. A coluna Produtos é expandida para uma linha por produto. O resultado é:

ID_Pedido	Data	Cliente	Produto	Quantidade	Categoria	Preço_Unitário	Endereço	Cidade	Distrito
P001	15/01/2024	João Silva	Computador	1	Eletrónica	1200.00	Rua 123	Sintra	Lisboa
P001	15/01/2024	João Silva	Impressora	1	Eletrónica	300.00	Rua 123	Sintra	Lisboa
P002	16/01/2024	Ana Sousa	Smartphone	2	Eletrónica	800.00	Rua 456	Gaia	Porto
P002	16/01/2024	Ana Sousa	Tablet	1	Eletrónica	400.00	Rua 456	Gaia	Porto
P003	16/01/2024	João Silva	Mesa Escritório	1	Móveis	150.00	Rua 123	Sintra	Lisboa
P004	17/01/2024	Maria Pereira	Cadeira Escritório	4	Móveis	75.00	Av. Central	Coimbra	Coimbra
P004	17/01/2024	Maria Pereira	Secretária	1	Móveis	200.00	Av. Central	Coimbra	Coimbra
P005	18/01/2024	Ana Sousa	Impressora	1	Eletrónica	300.00	Rua 456	Gaia	Porto

Segunda Forma Normal: Para atingir a 2FN, todos os atributos não chave devem depender funcionalmente de toda a chave primária e não de partes dela. Na tabela inicial, identificaram-se as seguintes dependências funcionais: `ID_Pedido` determina `Data` e `ID_Cliente`, enquanto o par `(ID_Pedido, nome_produto)` determina `quantidade_produto`, `categoria_produto` e `preco_unitario_produto`. Estas dependências demonstram que alguns atributos dependem exclusivamente de `ID_Pedido`, enquanto outros dependem do par `(ID_Pedido, nome_produto)`, violando o princípio da 2FN.

Para resolver isso, a tabela foi dividida em três relações. A tabela `tbl_pedidos` contém os dados relacionados ao pedido (`ID_Pedido`, `Data`, `ID_Cliente`), assegurando que todos os atributos dependem exclusivamente de `ID_Pedido`. A tabela `tbl_clientes` armazena informações dos clientes (`ID_Cliente`, `Cliente`, `Endereço`, `Cidade`, `Distrito`), garantindo que todos os atributos dependem de `ID_Cliente`. Por fim, a tabela `Itens_Pedido` associa os produtos aos respetivos pedidos (`ID_Pedido`, `nome_produto`, `quantidade_produto`, `categoria_produto`, `preco_unitario_produto`), de forma que cada atributo depende integralmente do par `(ID_Pedido, nome_produto)`. Esta separação

elimina dependências parciais, cumpre os requisitos da 2FN e organiza os dados de forma eficiente.

Tabela Itens_Pedido (Produtos por pedido):

ID_Pedido	nome_produto	quantidade_produto	categoria_produto	preço_unitário_produto
P001	Computador	1	Eletrónica	1200.00€
P001	Impressora	1	Eletrónica	300 €
P002	Smartphone	2	Eletrónica	800.00€
P002	Tablet	1	Eletrónica	400.00€
P003	Cadeira Escritório	4	Móveis	75.00€
P003	Secretária	1	Móveis	200.00€
P004	Impressora	1	Eletrónica	300 €

tbl_pedidos

ID_Pedido	Data	ID_Cliente
P001	15/01/2024	1
P002	16/01/2024	2
P003	16/01/2024	1
P004	17/01/2024	3
P005	18/01/2024	2

tbl_clientes

ID_Cliente	Cliente	Endereço	Cidade	Distrito
1	João Silva	Rua 123	Sintra	Lisboa
2	Ana Sousa	Rua 456	Gaia	Porto
3	Maria Pereira	Av. Central	Coimbra	Coimbra

Terceira Forma Normal: Para atingir a 3FN, é necessário eliminar todas as dependências transitivas, garantindo que cada atributo não chave dependa apenas e diretamente da chave primária da sua tabela. No caso da estrutura inicial, foi identificada uma dependência transitiva na tabela `tbl_clientes`, onde o atributo `Distrito` dependia de `Cidade`, e `Cidade`, por sua vez, dependia da chave primária `ID_Cliente`. Esta configuração viola os princípios da 3FN, pois `Distrito` não está diretamente relacionado com a chave primária, mas sim com um atributo não chave.

Para resolver esta situação, foi criada uma nova tabela chamada `Tabela Cidades`, que armazena as informações sobre cada cidade e o respetivo distrito. A tabela `Tabela Cidades` contém os atributos `Cidade` e `Distrito`, assegurando que cada cidade está associada unicamente ao seu distrito, eliminando a redundância e estabelecendo uma relação direta entre estes atributos. Simultaneamente, a tabela `tbl_clientes` foi ajustada para conter apenas os atributos diretamente relacionados ao cliente, nomeadamente `ID_Cliente`, `Cliente`, `Endereço` e `Cidade`. O atributo

Cidade agora serve como uma ligação à tabela Tabela Cidades, permitindo consultar o distrito correspondente sem violar a 3FN.

Com esta separação, as dependências transitivas foram eliminadas, e cada atributo não chave depende apenas da chave primária da sua tabela. Este processo garante uma estrutura normalizada, reduzindo redundâncias e assegurando a consistência e a integridade dos dados. Assim, as tabelas estão devidamente organizadas de acordo com os princípios da 3FN.

Tabela Itens_Pedido (Produtos por pedido):

ID_Pedido	nome_produto	quantidade_produto	categoria_produto	preço_unitário_produto
P001	Computador	1	Eletrónica	1200.00€
P001	Impressora	1	Eletrónica	300 €
P002	Smartphone	2	Eletrónica	800.00€
P002	Tablet	1	Eletrónica	400.00€
P003	Cadeira Escritório	4	Móveis	75.00€
P003	Secretária	1	Móveis	200.00€
P004	Impressora	1	Eletrónica	300 €

tbl_clientes

ID_Cliente	Cliente	Endereço	Cidade
1	João Silva	Rua 123	Sintra
2	Ana Sousa	Rua 456	Gaia
3	Maria Pereira	Av. Central	Coimbra

Tabela Cidades

Cidade	Distrito
Sintra	Lisboa
Gaia	Porto
Coimbra	Coimbra

tbl_pedidos

ID_Pedido	Data	ID_Cliente
P001	15/01/2024	1
P002	16/01/2024	2
P003	16/01/2024	1
P004	17/01/2024	3
P005	18/01/2024	2

4.a)

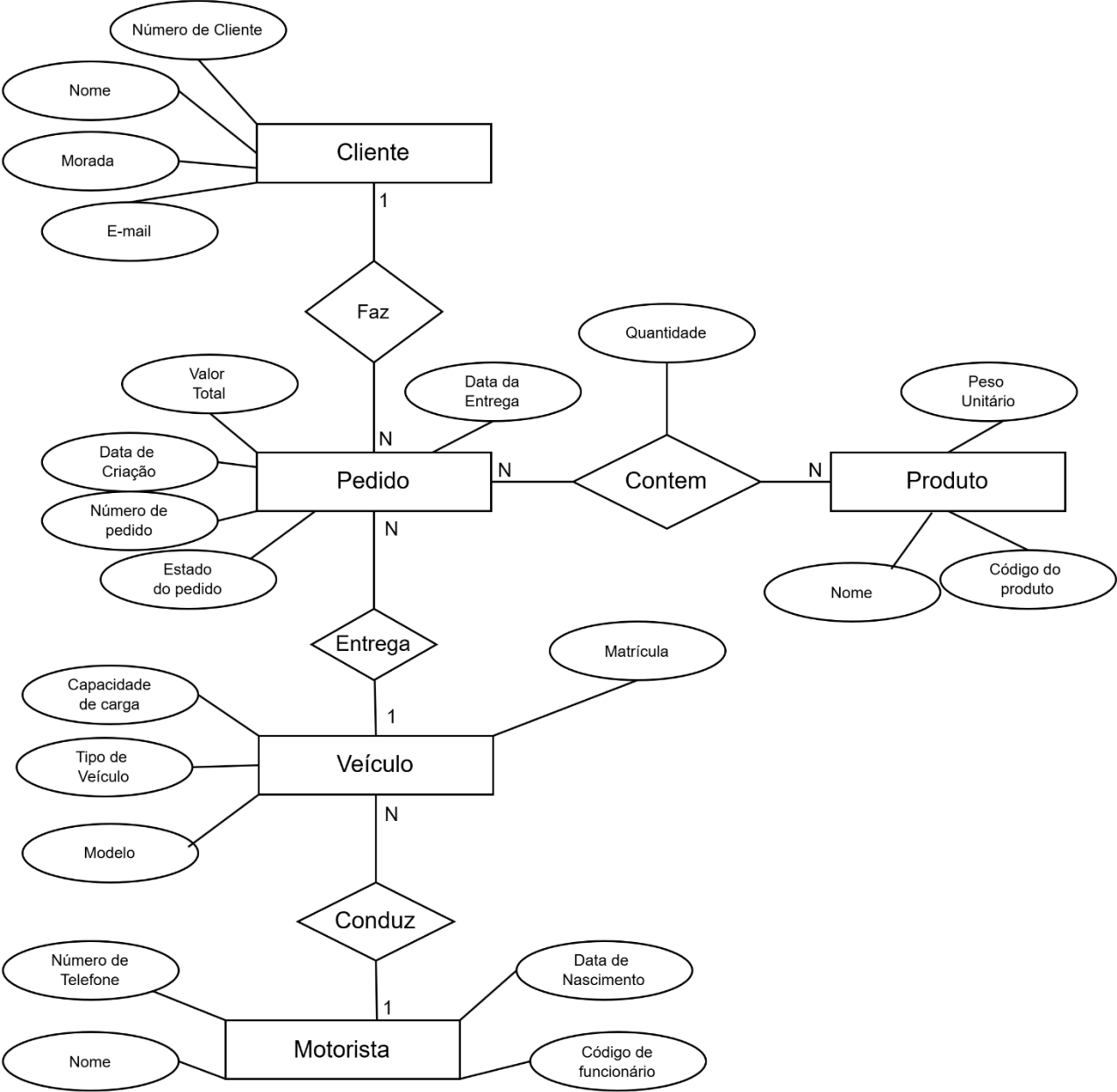
Tabela das Entidades e seus Atributos

Entidade	Atributos
Veículo	Matrícula , Modelo, Capacidade_Carga, Tipo
Motorista	Código_Funcionário, Nome, Data_Nascimento, Telefone
Cliente	Número_Cliente , Nome, Email, Morada
Pedido	Número_Pedido, Data_Criação, Valor_Total, Estado, Data_Entrega
Produto	Código_Produto , Nome_Produto, Peso_Unitário

Tabela de Relacionamentos e Cardinalidade

Relacionamento	Entidades Envolvidas	Cardinalidade	Descrição
Faz (Cliente-Pedido)	Cliente ↔ Pedido	1:N	Um cliente pode fazer vários pedidos, mas cada pedido é associado a um único cliente.
Contem (Pedido-Produto)	Pedido ↔ Produto	N:N	Um pedido pode incluir vários produtos, e um produto pode ser incluído em vários pedidos.
Entrega (Veículo-Pedido)	Veículo ↔ Pedido	1:N	Um veículo pode transportar vários pedidos, mas cada pedido é transportado por um único veículo.
Conduz (Motorista-Veículo)	Motorista ↔ Veículo	1:N	Um motorista pode conduzir vários veículos, mas cada veículo tem apenas um motorista.

Diagrama do Modelo Entidade-Relação:



4.b)

