

Kanta Íatima

ATACADO E VAREJO

PROJETO DE BANCO DE DADOS

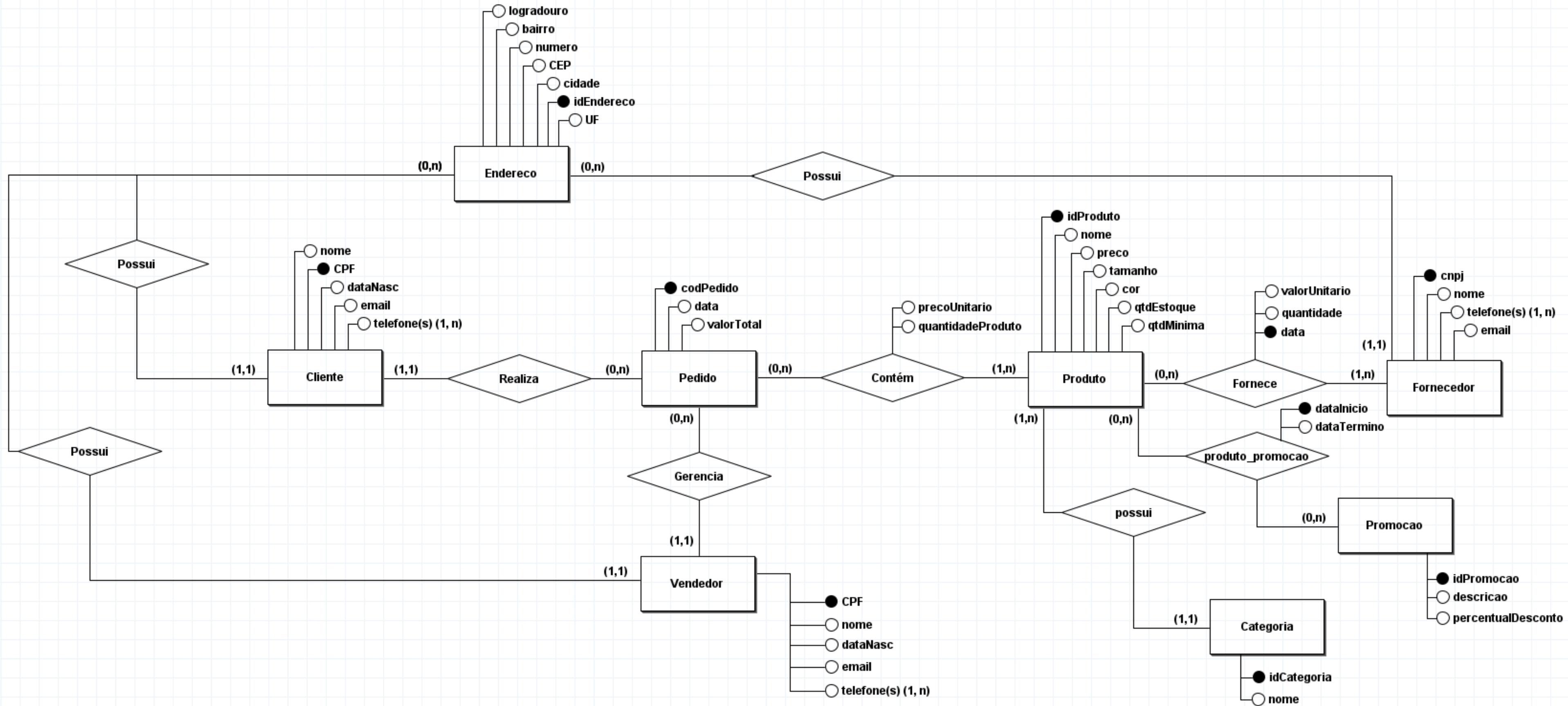
David Macedo, Geison, Igor, Wendel

Agenda

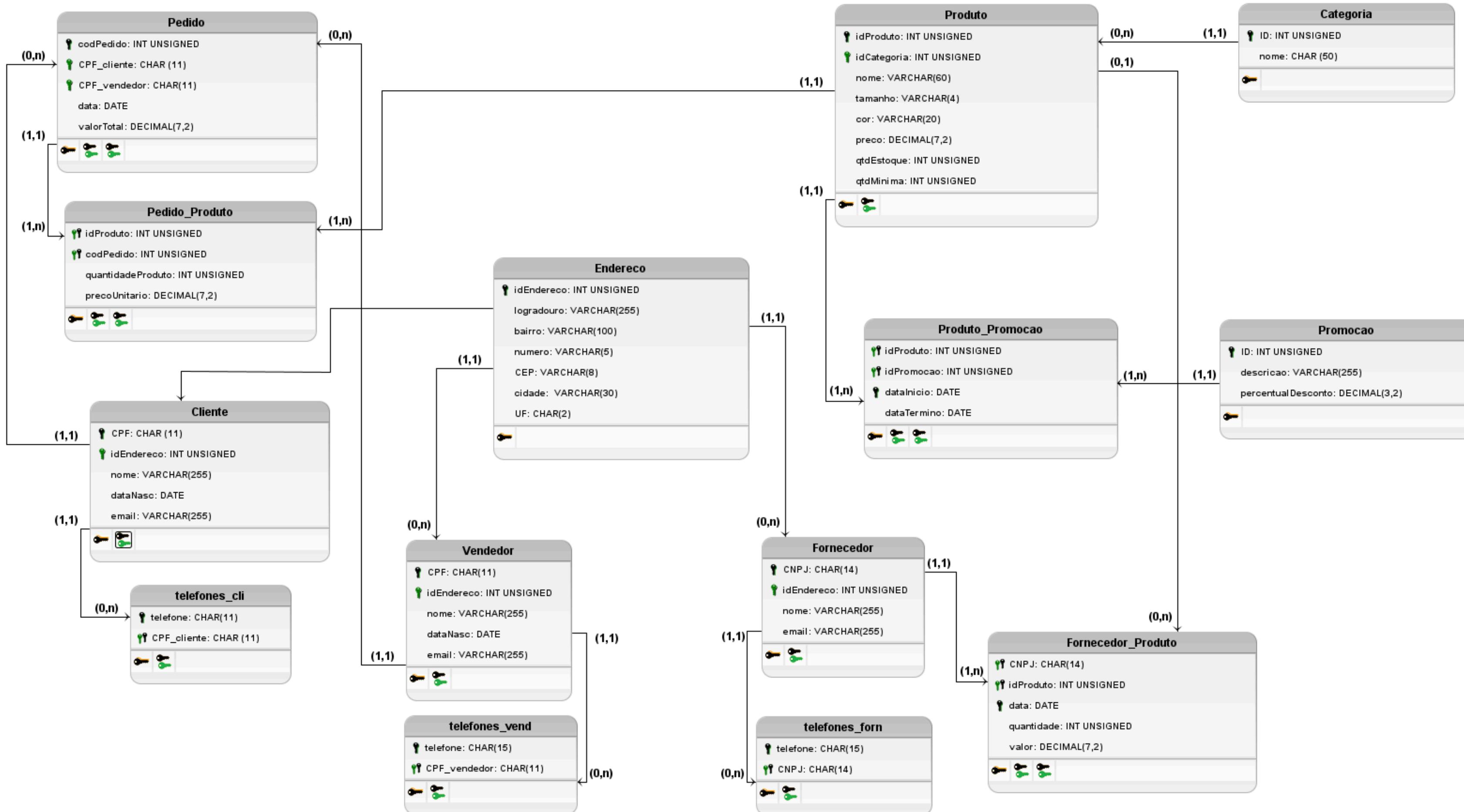
- MySQL - Esquema Conceitual
- MySQL - Esquema Lógico
- MySQL - Tabelas escolhidas para a mesclagem
- MongoDB - Modelo de dados
- MongoDB - Exemplo de dados
- Consultas
- Considerações finais



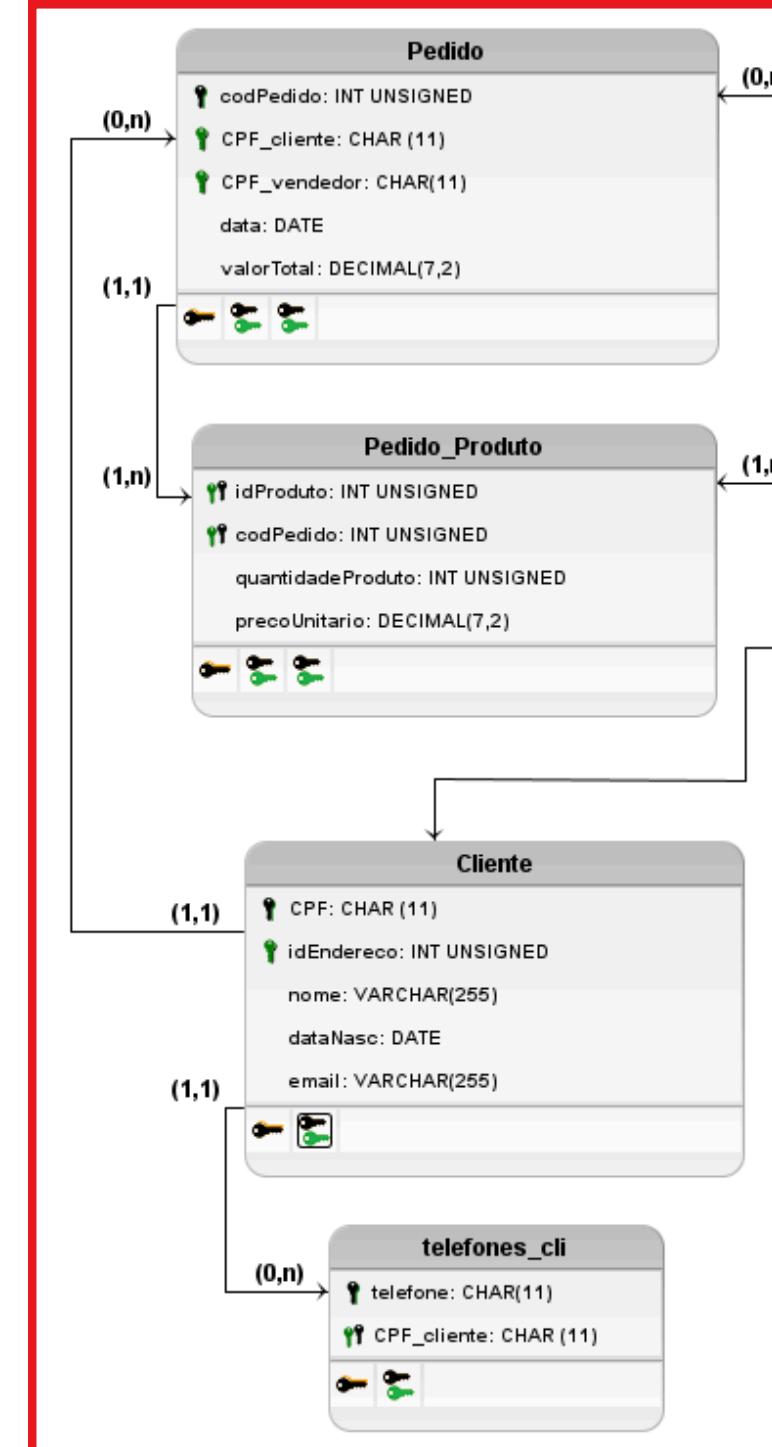
MySQL - Esquema Conceitual



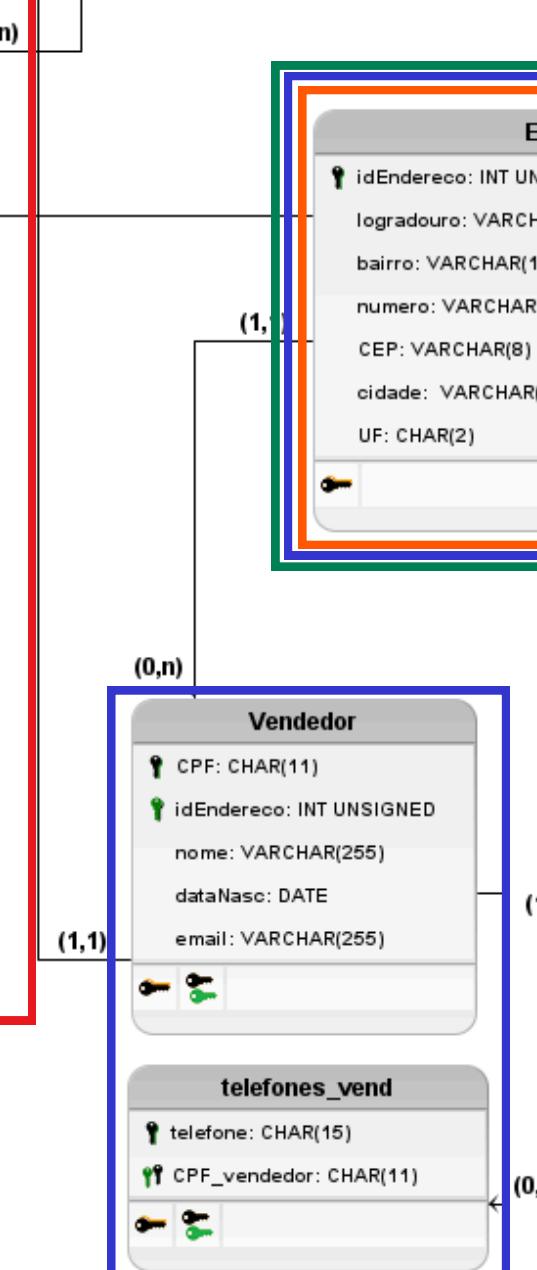
MySQL - Esquema Lógico



Modelo de documentos

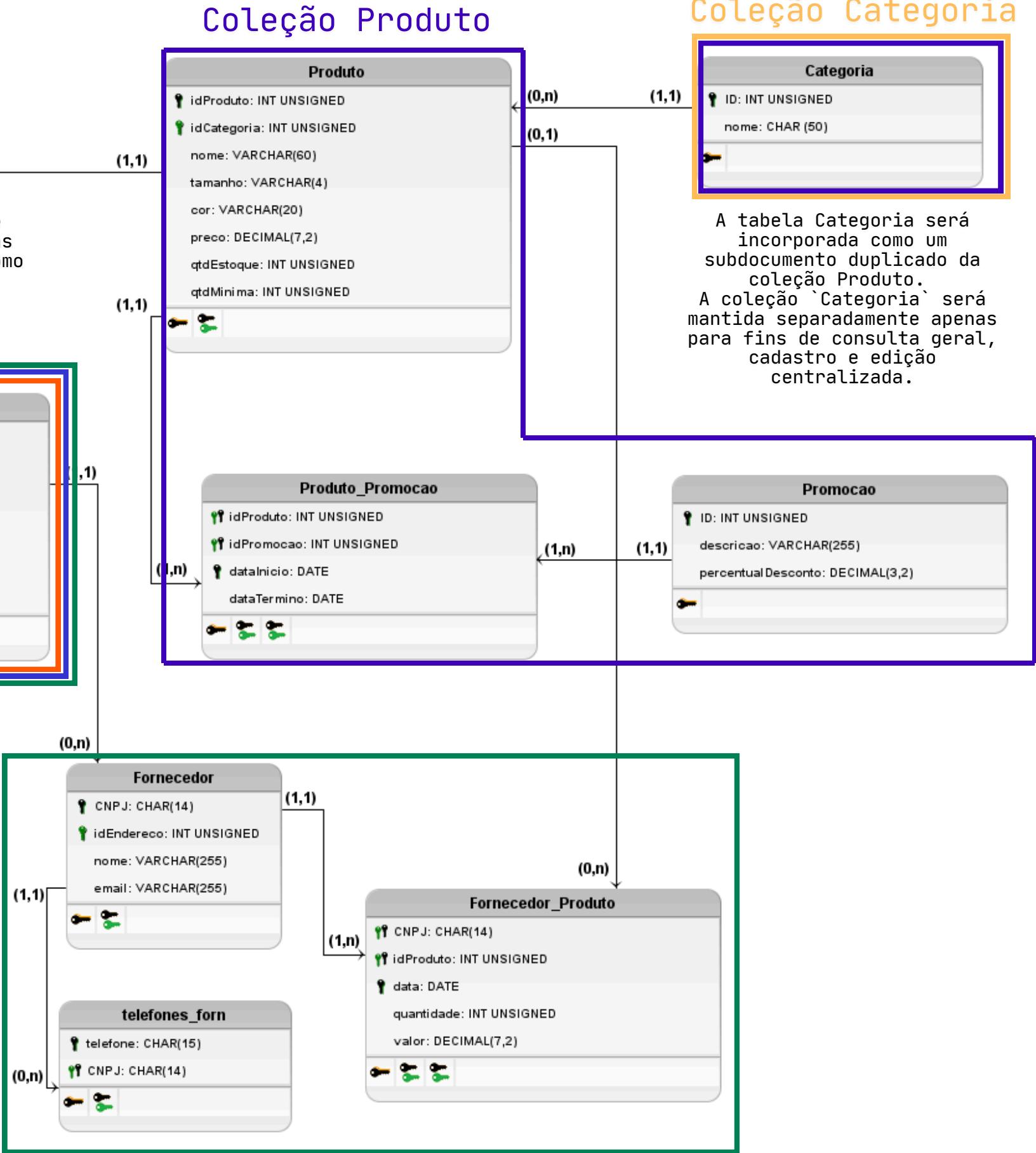


Coleção Cliente



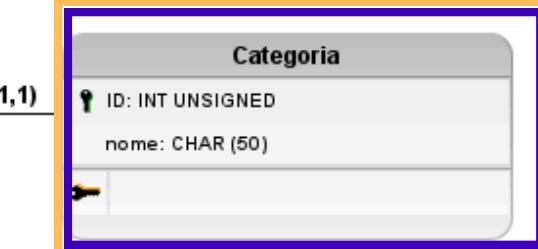
Coleção Vendedor

A tabela endereço irá se tornar um subdocumento das coleções que a incluem, como Fornecedor, Vendedor e Cliente



Coleção Fornecedor

Coleção Categoria



A tabela Categoria será incorporada como um subdocumento duplicado da coleção Produto. A coleção `Categoria` será mantida separadamente apenas para fins de consulta geral, cadastro e edição centralizada.

Modelo de documento com dados do JSON



Collections: Exemplos de dados

db.getCollection('vendedor').find({})						
	_id	nome	email	dataNasc	endereco	telefones
1	11122233344	Carlos Lima	carlos@email.com	1982-03-10	{ 6 fields }	[1 element]
2	22233344455	Marina Reis	marina@email.com	1990-04-04	{ 6 fields }	[1 element]
3	33344455566	Roberto Silva	roberto@email.com	1987-05-05	{ 6 fields }	[1 element]
4	44455566677	Larissa Souza	larissa@email.com	1983-06-06	{ 6 fields }	[1 element]
5	55566677788	Tiago Ramos	tiago@email.com	1995-07-07	{ 6 fields }	[1 element]
6	66677788899	Bruna Dias	bruna@email.com	1991-08-08	{ 6 fields }	[1 element]
7	77788899900	Marcelo ...	marcelo@email.com	1993-09-09	{ 6 fields }	[1 element]
8	88899900011	Patrícia ...	patricia@email.com	1989-10-10	{ 6 fields }	[1 element]
9	99900011122	Fernando ...	fernando@email.com	1994-11-11	{ 6 fields }	[1 element]
10	00011122233	Isabela ...	isabela@email.com	1996-12-12	{ 6 fields }	[1 element]

db.getCollection('cliente').find({})							
	_id	nome	email	pedidos	dataNasc	endereco	telefones
1	12345678901	João Silva	joao@email.com	[2 elements]	1990-05-15	{ 6 fields }	[2 elements]
2	98765432100	Maria Souza	maria@email.com	[2 elements]	1985-08-22	{ 6 fields }	[1 element]
3	11111111111	Carlos ...	carlos@email.com	[3 elements]	1992-02-10	{ 6 fields }	[1 element]
4	22222222222	Ana Paula	ana@email.com	[4 elements]	1980-11-30	{ 6 fields }	[1 element]
5	33333333333	Bruno Costa	bruno@email.com	[2 elements]	1975-07-18	{ 6 fields }	[1 element]
6	44444444444	Lucia Alves	lucia@email.com	[2 elements]	1999-09-09	{ 6 fields }	[1 element]
7	55555555555	Pedro Rocha	pedro@email.com	[2 elements]	2000-01-01	{ 6 fields }	[1 element]
8	66666666666	Sofia Martins	sofia@email.com	[2 elements]	1994-03-03	{ 6 fields }	[1 element]
9	77777777777	Marcos Lima	marcos@email.com	[2 elements]	1988-12-12	{ 6 fields }	[1 element]
10	88888888888	Juliana Faria	juliana@email.com	[2 elements]	1991-06-06	{ 6 fields }	[1 element]
11	11111111112	Ana Souza	ana@email.com	null	1990-05-20	{ 6 fields }	null
12	22222222224	João Pedro	joao@email.com	null	1988-09-15	{ 6 fields }	null
13	01234567890	Novo Clien...	novo.teste...	[1 element]	1990-01-01	{ 6 fields }	[1 element]

Consulta 1 Análise de Rentabilidade por Categoria de Produto

MySQL

```
● ● ● Consultas.json
SELECT
    c.nome AS Categoria,
    SUM(pp.quantidadeProduto) AS TotalItensVendidos,
    SUM(pp.quantidadeProduto * pp.precoUnitario) AS ReceitaBruta,
    SUM(pp.quantidadeProduto * Custo.custo_medio_unitario) AS CustoTotal,
    (SUM(pp.quantidadeProduto * pp.precoUnitario) -
     SUM(pp.quantidadeProduto * Custo.custo_medio_unitario)) AS LucroBruto
FROM Categoria c
JOIN Produto p ON c.ID = p.idCategoria
JOIN Pedido_Produto pp ON p.idProduto = pp.idProduto
JOIN (
    -- Subconsulta para calcular o custo médio de cada produto
    SELECT
        idProduto,
        AVG(valor) AS custo_medio_unitario
    FROM Fornecedor_Produto
    GROUP BY idProduto
) AS Custo ON p.idProduto = Custo.idProduto
GROUP BY c.nome
ORDER BY LucroBruto DESC;
```

MongoDB

```
● ● ● Consultas.json
db.cliente.aggregate([
    { $unwind: "$pedidos" },
    { $unwind: "$pedidos.itens" },
    { $group: {
        _id: "$pedidos.itens.idProduto",
        totalVendido: { $sum: "$pedidos.itens.quantidade" },
        receitaBruta: { $sum: { $multiply: ["$pedidos.itens.quantidade",
            "$pedidos.itens.precoUnitario"] } } },
    { $lookup: { from: "produto", localField: "_id", foreignField: "_id", as: "produto" } },
    { $unwind: "$produto" },
    { $lookup: {
        from: "fornecedor",
        let: { prodId: "$_id" },
        pipeline: [
            { $unwind: "$produtos_fornecidos" },
            { $match: { $expr: { $eq: ["$produtos_fornecidos.idProduto", "$$prodId"] } } },
            { $group: { _id: null, custoMedio: { $avg: "$produtos_fornecidos.valor" } } }
        ], as: "custo" },
    { $unwind: { path: "$custo", preserveNullAndEmptyArrays: true } },
    {$addFields: {
        custoMedio: { $ifNull: ["$custo.custoMedio", 0] },
        custoTotalProduto: { $multiply: ["$totalVendido", { $ifNull: ["$custo.custoMedio", 0] }] },
        categoriaNome: "$produto.categoria.nome" },
    {$group: {
        _id: "$categoriaNome",
        ReceitaBruta: { $sum: "$receitaBruta" },
        CustoTotal: { $sum: "$custoTotalProduto" },
        TotalItensVendidos: { $sum: "$totalVendido" } },
    { $addFields: { LucroBruto: { $subtract: ["$ReceitaBruta", "$CustoTotal"] }, Categoria: "$_id" } },
    { $project: { _id: 0,
        Categoria: "$Categoria", TotalItensVendidos: "$TotalItensVendidos",
        ReceitaBruta: { $round: ["$ReceitaBruta", 2] },
        CustoTotal: { $round: ["$CustoTotal", 2] },
        LucroBruto: { $round: ["$LucroBruto", 2] } } },
    { $sort: { LucroBruto: -1 } }
]);
```

Consulta 1 - Resultados

Lucro bruto por Categoria de Produto

MySQL

	Categoria	TotalItensVendidos	ReceitaBruta	CustoTotal	LucroBruto
▶	Lingerie	15	448.50	232.500000	216.000000
	Pijamas	6	329.40	145.000000	184.400000
	Camisolas	3	269.70	120.000000	149.700000
	Sutiãs	8	319.20	190.000000	129.200000
	Conjuntos	2	199.80	100.000000	99.800000
	Acessórios	2	179.80	90.000000	89.800000
	Robes	2	119.80	60.000000	59.800000
	Cuecas	4	91.70	32.000000	59.700000
	Boxers	5	104.60	52.000000	52.600000
	Meias	5	49.50	25.000000	24.500000

MongoDB

```
{  
  Categoria: 'Lingerie',  
  TotalItensVendidos: 15,  
  ReceitaBruta: 448.5,  
  CustoTotal: 232.5,  
  LucroBruto: 216  
},  
,  
{  
  Categoria: 'Pijamas',  
  TotalItensVendidos: 6,  
  ReceitaBruta: 329.4,  
  CustoTotal: 145,  
  LucroBruto: 184.4  
},  
,  
{  
  Categoria: 'Camisolas',  
  TotalItensVendidos: 3,  
  ReceitaBruta: 269.7,  
  CustoTotal: 120,  
  LucroBruto: 149.7  
},  
,  
{  
  Categoria: 'Sutiãs',  
  TotalItensVendidos: 8,  
  ReceitaBruta: 319.2,  
  CustoTotal: 190,  
  LucroBruto: 129.2  
},  
}
```

Consulta 2 Desempenho de Vendedores e Valor Médio por Pedido

MySQL

```
● ● ● Consultas.json
SELECT
    v.nome AS Vendedor,
    COUNT(p.codPedido) AS TotalPedidos,
    SUM(p.valorTotal) AS FaturamentoTotal,
    AVG(p.valorTotal) AS TicketMedio
FROM Vendedor v
JOIN Pedido p ON v.CPF = p.CPF_vendedor
GROUP BY v.CPF, v.nome
ORDER BY FaturamentoTotal DESC;
```

MongoDB

```
CONSULTA 2
db.vendedor.aggregate([
    { $lookup: {
        from: "cliente",
        let: { vendedorCPF: "$_id" },
        pipeline: [
            { $match: { pedidos: { $ne: null, $exists: true, $not: { $size: 0 } } } },
            { $unwind: "$pedidos" },
            { $match: { $expr: { $eq: ["$pedidos.CPF_vendedor", "$$vendedorCPF"] } } },
            { $replaceRoot: { newRoot: "$pedidos" } }
        ],
        as: "pedidos"
    }, {
        // Cria os campos agregados
        $addFields: {
            TotalPedidos: { $size: "$pedidos" },
            FaturamentoTotal: { $sum: "$pedidos.valorTotal" },
            TicketMedio: {
                $cond: [
                    { $gt: [{ $size: "$pedidos" }, 0] },
                    { $avg: "$pedidos.valorTotal" },
                    0
                ]
            }
        }
    },
    // Seleciona os campos desejados
    $project: { _id: 0,
        Vendedor: "$nome",
        TotalPedidos: "$TotalPedidos",
        FaturamentoTotal: { $round: ["$FaturamentoTotal", 2] },
        TicketMedio: { $round: ["$TicketMedio", 2] } },
        { $sort: { FaturamentoTotal: -1 } }
    ]);
});
```

Consulta 2 - Resultados

Desempenho de Vendedores e Valor Médio por Pedido

MySQL

	Vendedor	TotalPedidos	FaturamentoTotal	TicketMedio
▶	Carlos Lima	7	1064.00	152.000000
	Fernando Campos	2	379.80	189.900000
	Bruna Dias	2	239.90	119.950000
	Patrícia Nogueira	2	199.90	99.950000
	Tiago Ramos	2	184.80	92.400000
	Roberto Silva	2	179.90	89.950000
	Larissa Souza	2	159.80	79.900000
	Marina Reis	2	125.40	62.700000
	Marcelo Alves	2	115.00	57.500000
	Isabela Freitas	1	49.90	49.900000

MongoDB

```
{  
    Vendedor: 'Carlos Lima',  
    TotalPedidos: 7,  
    FaturamentoTotal: 1064,  
    TicketMedio: 152  
},  
{  
    Vendedor: 'Fernando Campos',  
    TotalPedidos: 2,  
    FaturamentoTotal: 379.8,  
    TicketMedio: 189.9  
},  
{  
    Vendedor: 'Bruna Dias',  
    TotalPedidos: 2,  
    FaturamentoTotal: 239.9,  
    TicketMedio: 119.95  
},  
{  
    Vendedor: 'Patrícia Nogueira',  
    TotalPedidos: 2,  
    FaturamentoTotal: 199.9,  
    TicketMedio: 99.95  
},
```

Consulta 3 Análise de Clientes que Compram Itens Promocionais

MySQL



Consultas.json

SELECT

```
c.nome AS Cliente,  
c.email,  
COUNT(DISTINCT p.idProduto) AS ItensPromocionaisDistintosComprados  
FROM Cliente c  
JOIN Pedido ped ON c.CPF = ped.CPF_cliente  
JOIN Pedido_Produto pp ON ped.codPedido = pp.codPedido  
JOIN Produto p ON pp.idProduto = p.idProduto  
WHERE p.idPromocao IS NOT NULL  
GROUP BY c.CPF, c.nome, c.email  
ORDER BY ItensPromocionaisDistintosComprados DESC;
```

MongoDB

CONSULTA 3

```
db.cliente.aggregate([  
    // Filtra clientes que têm pedidos  
    { $match: { pedidos: { $ne: null, $exists: true, $not: { $size: 0 } } } },  
    { $unwind: "$pedidos" },  
    { $unwind: "$pedidos.itens" },  
    // JOIN com produtos (para verificar promoção)  
    { $lookup: {  
        from: "produto",  
        localField: "pedidos.itens.idProduto",  
        foreignField: "_id",  
        as: "produto" } },  
    { $unwind: "$produto" },  
    // Filtra apenas produtos com promoção  
    { $match: {  
        "produto.promocoes": { $ne: null, $exists: true, $not: { $size: 0 } } } },  
    // Agrupa por cliente e coleta os produtos distintos com promoção  
    { $group: { _id: {  
        CPF: "$_id",  
        nome: "$nome",  
        email: "$email" },  
        produtosPromocionais: { $addToSet: "$produto._id" } } },  
    // Conta os produtos distintos comprados com promoção  
    {  
        $project: {  
            Cliente: "$_id.nome",  
            Email: "$_id.email",  
            ItensPromocionaisDistintosComprados: { $size: "$produtosPromocionais" },  
            _id: 0 } },  
        { $sort: { ItensPromocionaisDistintosComprados: -1 } }  
]);
```

Consulta 3 - Resultados

Análise de Clientes que Compram Itens Promocionais

MySQL

	Cliente	email	ItensPromocionaisDistintosComprados
▶	Ana Paula	ana@email.com	4
	Novo Cliente Teste	novo.teste@email.com	3
	Carlos Andrade	carlos@email.com	3
	João Silva	joao@email.com	2
	Bruno Costa	bruno@email.com	2
	Lucia Alves	lucia@email.com	2
	Pedro Rocha	pedro@email.com	2
	Sofia Martins	sofia@email.com	2
	Marcos Lima	marcos@email.com	2
	Juliana Faria	juliana@email.com	2
	Maria Souza	maria@email.com	1

MongoDB

```
{  
  Cliente: 'Ana Paula',  
  email: 'ana@email.com',  
  ItensPromocionaisDistintosComprados: 4  
},  
{  
  Cliente: 'Novo Cliente Teste',  
  email: 'novo.teste@email.com',  
  ItensPromocionaisDistintosComprados: 3  
},  
{  
  Cliente: 'Carlos Andrade',  
  email: 'carlos@email.com',  
  ItensPromocionaisDistintosComprados: 3  
},
```

Consulta 4 Análise de Estoque Crítico (Produtos Abaixo do Mínimo)

MySQL

Consultas.json

```
SELECT
    p.idProduto,
    p.nome,
    p.qtdEstoque,
    p.qtdMinima,
    (p.qtdMinima - p.qtdEstoque) AS UnidadesFaltantes
FROM Produto p
WHERE
    p.qtdEstoque < p.qtdMinima AND
    p.idProduto NOT IN (
        -- Subconsulta para encontrar produtos em promoções ativas
        SELECT p2.idProduto
        FROM Produto p2
        JOIN Promocao promo ON p2.idPromocao = promo.ID
        WHERE CURDATE() BETWEEN promo.dataInicio AND promo.dataTermino
    );
```

MongoDB

```
CONSULTA 4
db.produto.aggregate([
    { $match: { $expr: { $lt: ["$qtdEstoque", "$qtdMinima"] } } },
    // Filtra produtos SEM promoção ativa
    { $match: {
        $or: [
            { promocoes: null },
            { promocoes: { $exists: false } },
            { promocoes: { $size: 0 } },
            {
                $expr: { $not: { $anyElementTrue: {
                    $map: {
                        input: "$promocoes",
                        as: "promo",
                        in: {
                            $and: [
                                { $gte: [new Date(), { $dateFromString: { dateString: "$$promo.dataInicio" } }] },
                                { $lte: [new Date(), { $dateFromString: { dateString: "$$promo.dataTermino" } }] }
                            ] } } } } } ]
            },
        // Calcula unidades faltantes e projeta campos
        { $addFields: {
            UnidadesFaltantes: { $subtract: ["$qtdMinima", "$qtdEstoque"] } }
        },
        { $project: { _id: 0,
            idProduto: "$_id",
            nome: "$nome",
            qtdEstoque: "$qtdEstoque",
            qtdMinima: "$qtdMinima",
            UnidadesFaltantes: { $subtract: [
                "$qtdMinima",
                "$qtdEstoque" ] } } },
        { $sort: { UnidadesFaltantes: 1 } }
    ]})
```

Consulta 4 - Resultados

Análise de Estoque Crítico (Produtos Abaixo do Mínimo)

MySQL

	idProduto	nome	qtdEstoque	qtdMinima	UnidadesFaltantes
▶	15	Meia Esportiva	5	10	5
	16	Top Cropped	7	15	8

MongoDB

```
{  
  idProduto: 15,  
  nome: 'Meia Esportiva',  
  qtdEstoque: 5,  
  qtdMinima: 10,  
  UnidadesFaltantes: 5  
},  
{  
  idProduto: 16,  
  nome: 'Top Cropped',  
  qtdEstoque: 7,  
  qtdMinima: 15,  
  UnidadesFaltantes: 8  
}
```

Consulta 5 Ranking de Produtos Mais Vendidos por Período

MySQL

```
SELECT
    p.idProduto,
    p.nome AS produto,
    SUM(pp.quantidadeProduto) AS total_vendido
FROM Pedido_Produto pp
JOIN Pedido pe ON pp.codPedido = pe.codPedido
JOIN Produto p ON pp.idProduto = p.idProduto
WHERE pe.data BETWEEN '2025-07-01' AND '2025-07-10'
GROUP BY p.idProduto, p.nome
ORDER BY total_vendido DESC;
```

MongoDB

```
CONSULTA 5

db.cliente.aggregate([
    // Filtrar clientes que possuem pedidos
    {
        $match: {
            pedidos: { $ne: null, $exists: true, $not: { $size: 0 } }
        }
    },
    { $unwind: "$pedidos" },
    { $match: {
        "pedidos.data": {
            $gte: "2025-07-01",
            $lte: "2025-07-10"
        }
    } },
    { $unwind: "$pedidos.itens" },
    { $lookup: {
        from: "produto",
        localField: "pedidos.itens.idProduto",
        foreignField: "_id",
        as: "produtoInfo" }
    },
    { $unwind: "$produtoInfo" },
    { $group: { _id: {
        idProduto: "$pedidos.itens.idProduto",
        nome: "$produtoInfo.nome"
    }},
        total_vendido: { $sum: "$pedidos.itens.quantidade" } }
    },
    { $project: { _id: 0,
        idProduto: "$_id.idProduto",
        produto: "$_id.nome",
        total_vendido: "$total_vendido"
    } },{ $sort: { total_vendido: -1 } }
]);
```

Consulta 5 - Resultados

Ranking de Produtos Mais Vendidos por Período

MySQL

	idProduto	produto	total_vendido
▶	1	Calcinha Renda	7
	2	Pijama Algodão	4
	5	Sutiã Comfort	3
	3	Camisola Seda	2
	4	Cueca Boxer	2
	6	Cueca Slip	2
	9	Meia Calça	2
	7	Robe Cetim	1
	8	Conjunto Luxo	1
	10	Cinta Modeladora	1

MongoDB

```
[  
  { idProduto: 1, produto: 'Calcinha Renda', total_vendido: 7 },  
  { idProduto: 2, produto: 'Pijama Algodão', total_vendido: 4 },  
  { idProduto: 5, produto: 'Sutiã Comfort', total_vendido: 3 },  
  { idProduto: 6, produto: 'Cueca Slip', total_vendido: 2 },  
  { idProduto: 3, produto: 'Camisola Seda', total_vendido: 2 },  
  { idProduto: 9, produto: 'Meia Calça', total_vendido: 2 },  
  { idProduto: 4, produto: 'Cueca Boxer', total_vendido: 2 },  
  { idProduto: 7, produto: 'Robe Cetim', total_vendido: 1 },  
  { idProduto: 10, produto: 'Cinta Modeladora', total_vendido: 1 },  
  { idProduto: 8, produto: 'Conjunto Luxo', total_vendido: 1 }  
]
```

CONSIDERAÇÕES FINAIS

- Diferenças entre os modelos
- Desafios Enfrentados
- Igualdade entre os bancos
- Vantagens Obtidas com o MongoDB

Material Consultado e Links Úteis

- MongoDB Community

<https://www.mongodb.com/try/download/community>

- Database References

<https://www.mongodb.com/docs/manual/reference/database-references/>

- Data Modeling Introduction

<https://www.mongodb.com/docs/manual/reference/database-references/>

- IDE Robo 3T

<https://blog.robomongo.org/studio3t-free/>

Atribuições

David

- Elaboração do slide
- Elaboração das consultas mongoDB

Geison

- Elaboração dos modelos de documentos e organização de coleções
- Geração de um mapeamento inicial de entidades para documentos MongoDB
(Elaboração de consultas de migração e povoamento).

Igor

- Consultas no MySQL
- Elaboração das consultas mongoDB

Wendel

- Elaboração dos modelos de documentos e organização de coleções
- Geração de um mapeamento inicial de entidades para documentos MongoDB
(Elaboração de consultas de migração e povoamento).

OBRIGADO!