

☐ Eng. Ambiental ☐ Eng. Civil ☒ Eng. da Computação ☐ Eng. Elétrica ☐ Eng. de Produção

Aluno (a): Lucas Pinheiro Santos RA: 5157075 Turma: _____

Valor: 5 Pontos Nota: _____ Data: _____

Data: 21/06/2023 Vista: _____ Nota final: _____

INSTRUÇÕES

1. Este trabalho deverá ser resolvido individualmente.
2. Faça o trabalho e envie **no diário de bordo** um PDF com os *prints completos* do código fonte e das telas de cada exercício. Enviar no diário de bordo da disciplina para o professor ou entregar impresso ao professor.
3. Apenas o exercício de desafio não será avaliado.
4. A leitura e interpretação das questões fazem parte do trabalho.

Bom trabalho!

Matéria: Aulas Práticas

I. Utilizando o Diagrama de Entidade e Relacionamento da Figura 1, realize as consultas solicitadas, utilizando a linguagem SQL.

- a. Crie uma procedure que deve buscar as subcategorias dos departamentos 1 e 2, abrir um cursor e excluir todas as subcategorias que não possuem produtos.

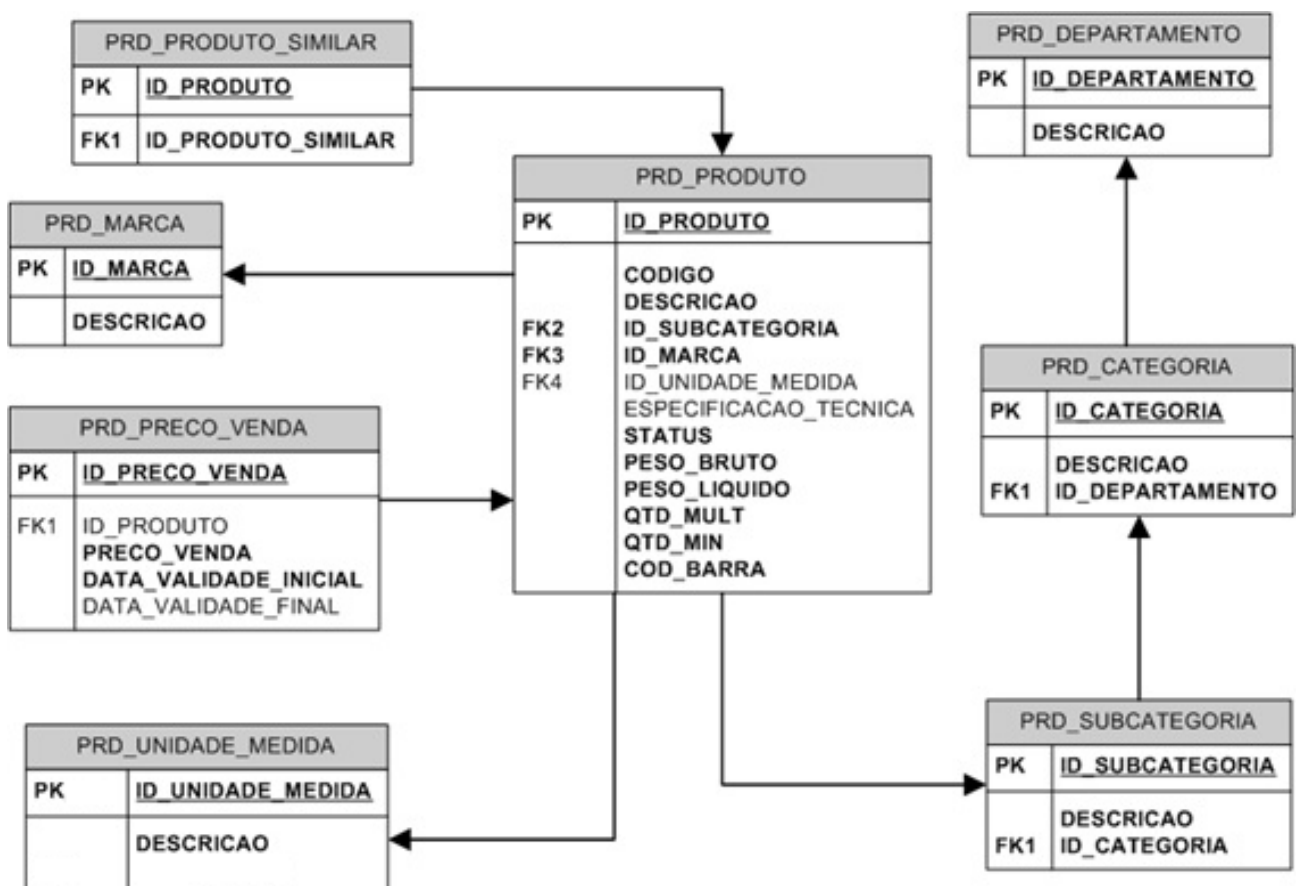


Figura 1 DER de um sistema compras. Adaptado de: <<https://www.devmedia.com.br/conceitos-de-mer-e-der-sql-server-2008-parte-1/18383>>. Acessado em: 06 nov. 2018.

```
CREATE PROCEDURE dbo.ProcedureExcluirSubcategoriasSemProdutos
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @Departamento1 INT, @Departamento2 INT;
    SET @Departamento1 = 1;
    SET @Departamento2 = 2;

    CREATE TABLE #TempSubcategorias (
        [ID_SUBCATEGORIA] INT
    );

    INSERT INTO #TempSubcategorias ([ID_SUBCATEGORIA])
    SELECT [ID_SUBCATEGORIA]
    FROM [dbo].[PRD_SUBCATEGORIA]
    WHERE [ID_CATEGORIA] IN (
        SELECT [ID_CATEGORIA]
        FROM [dbo].[PRD_CATEGORIA]
        WHERE [ID_DEPARTAMENTENTO] IN (@Departamento1, @Departamento2)
    );

    DECLARE @SubcategoriaID INT;

    DECLARE subcategoria_cursor CURSOR FOR
    SELECT [ID_SUBCATEGORIA]
    FROM #TempSubcategorias;

    OPEN subcategoria_cursor;

    FETCH NEXT FROM subcategoria_cursor INTO @SubcategoriaID;

    WHILE @@FETCH_STATUS = 0
    BEGIN
        -- Verificar se a subcategoria possui produtos
        IF NOT EXISTS (
            SELECT 1
            FROM [dbo].[PRD_PRODUTO]
```

```

WHERE [ID_SUBCATEGORIA] = @SubcategoriaID
)
BEGIN
    -- Excluir a subcategoria caso não possua produtos
    DELETE FROM [dbo].[PRD_SUBCATEGORIA]
    WHERE [ID_SUBCATEGORIA] = @SubcategoriaID;
END;

FETCH NEXT FROM subcategoria_cursor INTO @SubcategoriaID;
END;

CLOSE subcategoria_cursor;
DEALLOCATE subcategoria_cursor;

-- Remover a tabela temporária
DROP TABLE #TempSubcategorias;
END;

```

2. Utilizando o Diagrama de Entidade e Relacionamento da Figura 2, realize as consultas solicitadas, utilizando a linguagem SQL.

a. Faça a criação das estruturas DDL de todas as tabelas do DER.

```

CREATE TABLE [dbo].[canteiro](
[canteiroId] [int] IDENTITY(1,1) NOT NULL,[nome]
[char](20) NOT NULL,
[luzdiaria] [float] NOT NULL, [aguadiaria]
[float] NOT NULL,
PRIMARY KEY
CLUSTERED(
[canteiroId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON
[PRIMARY]GO
/***** Object: Table [dbo].[plantio]
Script Date: 26/03/2023 22:50:46 *****/SET
ANSI_NULLS ON
GO
SET
QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[plantio](
[plantioId] [int] IDENTITY(1,1) NOT NULL,
[plantaId] [int] NOT NULL, [funcId] [int]
NOT NULL, [canteiroId] [int] NOT
NULL,[data] [date] NOT NULL,
[sementes] [int] NOT NULL,
PRIMARY KEY
CLUSTERED(
[plantioId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,

```

ALLOW_ROW_LOCKS =ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]

) ON

[PRIMARY]GO

/***** Object: Table [dbo].[colhido]

Script Date: 26/03/2023 22:50:46 *****/SET

ANSI_NULLS ON

GO

SET

QUOTED_IDENTIFIER

ON

CREATE TABLE [dbo].[colhido](

[colhidoId] [int] IDENTITY(1,1) NOT NULL,

[plantaId] [int] NOT NULL, [funcId]

[int] NOT NULL, [canteiroId] [int]

NOT NULL, [data] [date] NOT

NULL, [quantidade] [int] NOT

NULL, [peso] [decimal](4, 3) NOT

NULL,

PRIMARY KEY

CLUSTERED(

[colhidoId] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,

ALLOW_ROW_LOCKS =ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)

ON [PRIMARY]

) ON

[PRIMARY]

GO

/***** Object: Table [dbo].[funcionario]

Script Date: 26/03/2023 22:50:46 *****/SET

ANSI_NULLS ON

GO

SET

QUOTED_IDENTIFIER

ON

CREATE TABLE [dbo].[funcionario](

[funcId] [int] IDENTITY(1,1) NOT NULL,

[nome] [char](80) NOT NULL,

[idade] [smallint] NOT NULL,

PRIMARY KEY

CLUSTERED(

[funcId] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,

ALLOW_ROW_LOCKS =ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)

ON [PRIMARY]

) ON

[PRIMARY]

GO

/***** Object: Table [dbo].[planta]

Script Date: 26/03/2023 22:50:46 *****/SET

ANSI_NULLS ON

GO

SET

QUOTED_IDENTIFIER

ON

CREATE TABLE [dbo].[planta](

[id] [int] NOT NULL,

[nome] [char](1) NOT NULL, [luzdiaria]

[decimal](4, 3) NOT NULL, [agua]

[decimal](4, 3) NOT NULL, [peso]

[decimal](4, 3) NOT NULL,

CONSTRAINT [PK_planta] PRIMARY KEY

CLUSTERED(

[id] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,

ALLOW_ROW_LOCKS =ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)

ON [PRIMARY]

) ON [PRIMARY]
GO

b. Faça a criação das estruturas DDL de todos os relacionamentos do DER.

```
ALTER TABLE [dbo].[colhido] WITH CHECK ADD CONSTRAINT [REL_colhi_canteiro] FOREIGN
KEY([canteiroId])
REFERENCES [dbo].[canteiro] ([canteiroId])GO
ALTER TABLE [dbo].[colhido] CHECK CONSTRAINT [REL_colhi_canteiro]
GO
ALTER TABLE [dbo].[colhido] WITH CHECK ADD CONSTRAINT [REL_colhi_funcio] FOREIGN
KEY([funcId])
REFERENCES [dbo].[funcionario] ([funcId])GO
ALTER TABLE [dbo].[colhido] CHECK CONSTRAINT [REL_colhi_funcio]
GO
ALTER TABLE [dbo].[colhido] WITH CHECK ADD CONSTRAINT [REL_colhi_planta] FOREIGN
KEY([plantaId])
REFERENCES [dbo].[planta] ([id])
GO
ALTER TABLE [dbo].[colhido] CHECK CONSTRAINT [REL_colhi_planta]
GO
ALTER TABLE [dbo].[plantio] WITH CHECK ADD CONSTRAINT [REL_plantio_canteiro] FOREIGN
KEY([canteiroId])
REFERENCES [dbo].[canteiro] ([canteiroId])GO
ALTER TABLE [dbo].[plantio] CHECK CONSTRAINT [REL_plantio_canteiro]
GO
ALTER TABLE [dbo].[plantio] WITH CHECK ADD CONSTRAINT [REL_plantio_funcionario] FOREIGN
KEY([funcId])
REFERENCES [dbo].[funcionario] ([funcId])GO
ALTER TABLE [dbo].[plantio] CHECK CONSTRAINT [REL_plantio_funcionario]
GO
ALTER TABLE [dbo].[plantio] WITH CHECK ADD CONSTRAINT [REL_plantio_planta] FOREIGN
KEY([plantaId])
REFERENCES [dbo].[planta] ([id])
GO
ALTER TABLE [dbo].[plantio] CHECK CONSTRAINT [REL_plantio_planta]
GO
USE
[master]GO
```

c. Faça a pesquisa de todos os plantios cuja data é maior que 2017.

```
SELECT * FROM [dbo].[plantio]
WHERE [data] > '2017-01-01';
```

d. Faça uma pesquisa de todos os funcionários cuja idade seja superior a 60 anos ou inferior a 18 anos.

```
SELECT *
FROM [dbo].[funcionario]
WHERE [idade] > 60 OR [idade] < 18;
```

e. Faça *procedure* que retorne tudo o que foi colhido por um determinado funcionário. Passe como parâmetro do procedimento uma variável correspondente à chave primária do funcionário.

```
IF OBJECT_ID('dbo.ObterColheitasPorFuncionario', 'P') IS NOT NULL
DROP PROCEDURE dbo.ObterColheitasPorFuncionario
GO
```

```
CREATE PROCEDURE dbo.ObterColheitasPorFuncionario
    @funcionarioId INT
AS
BEGIN
    SELECT *
    FROM [dbo].[colhido]
    WHERE [funcId] = @funcionarioId;
END
GO
```

f. Faça uma *view* que retorne os canteiros que possuem diversificação de plantios de plantas.

```
IF OBJECT_ID('dbo.ViewCanteirosDiversificados', 'V') IS NOT NULL
    DROP VIEW dbo.ViewCanteirosDiversificados
GO
```

```
CREATE VIEW dbo.ViewCanteirosDiversificados
AS
SELECT [canteiroId]
FROM [dbo].[plantio]
GROUP BY [canteiroId]
HAVING COUNT(DISTINCT [plantaId]) > 1;
```

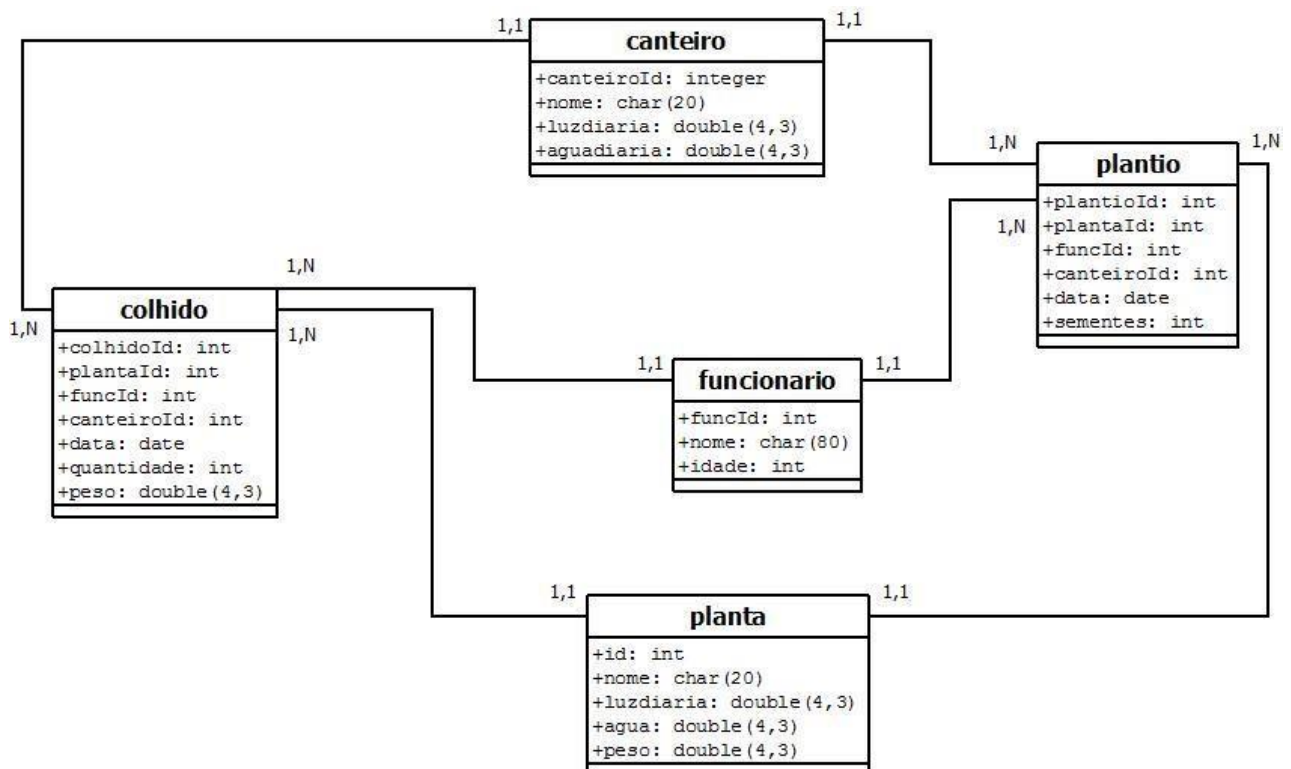


Figura 2 DER de uma plantação. Disponível em: < <http://eufacoprogramas.com/exemplos-de-consulta-e-criacao-de-tabelas-em-sql/>>. Acessado em: 06 nov. 2017.

Matéria: Aulas Teóricas

3. O que é um *Data Warehouse*? Qual diferença dos bancos de dados tradicionais e bancos de dados multidimensionais?

Um Data Warehouse é um sistema de armazenamento centralizado e organizado de dados que são extraídos, transformados e carregados (ETL) a partir de diferentes fontes de dados operacionais. Ele é projetado para suportar análises complexas e consultas ad-hoc, fornecendo uma visão histórica e consolidada dos dados de uma organização. A diferença dos bancos de dados tradicionais está na sua estruturação e modelagem, com um foco na otimização de consultas analíticas. Já os bancos de dados multidimensionais são projetados especificamente para análise de dados multidimensionais, utilizando um modelo dimensional que organiza os dados em torno de dimensões e medidas, facilitando análises OLAP (Online Analytical Processing) e a construção de cubos de dados.

4. Os bancos de dados tradicionais dão apoio ao processamento *on-line* de transações (*online transaction processing – OLTP*), com inclusões, atualizações e exclusões frequentes dos dados. Qual suporte é dado pelo *Data Warehouse* e para quais tipos de sistemas/tecnologias?

Os Data Warehouses fornecem suporte para o processamento analítico online (online analytical processing - OLAP), que se concentra em consultas complexas e análise de dados. Eles são projetados para oferecer uma visão consolidada e histórica dos dados, permitindo análises avançadas e a tomada de decisões estratégicas. O Data Warehouse é utilizado por sistemas e tecnologias que requerem análises e relatórios de dados, como inteligência de negócios (BI), ferramentas de visualização de dados, modelagem estatística e outras aplicações analíticas.

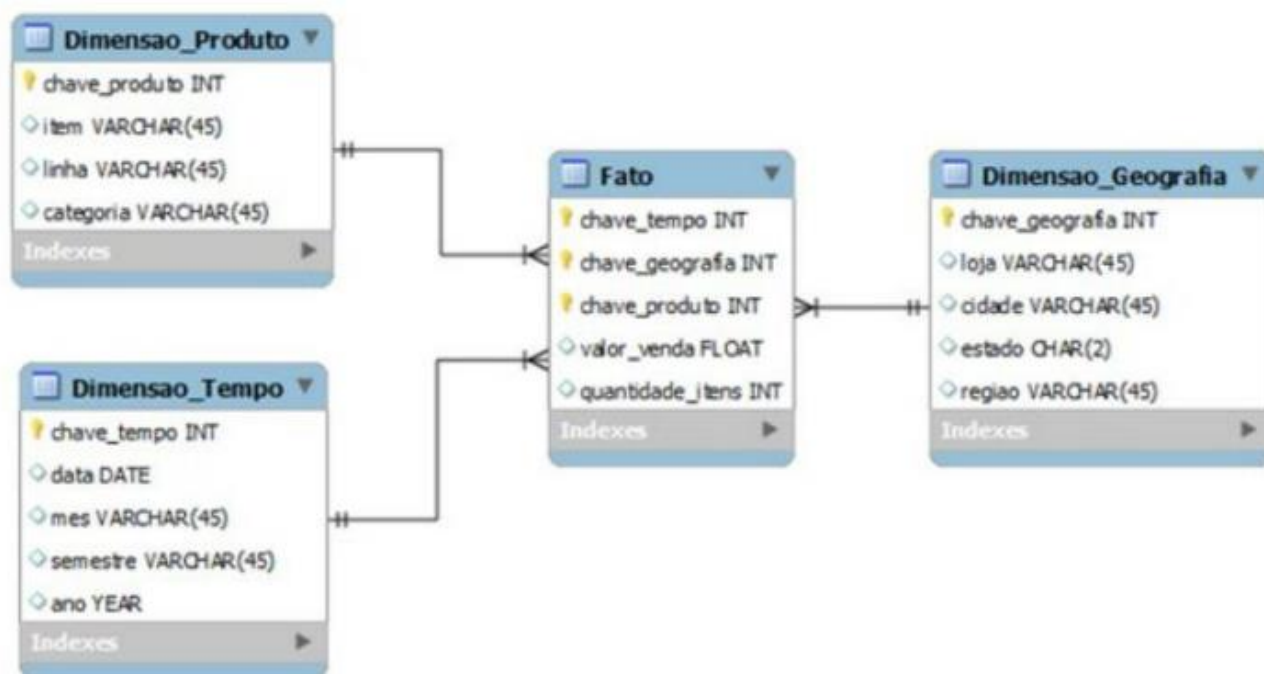
5. O que é um cubo, fato e dimensões nos bancos de dados multidimensionais? Qual a sua função?

Em bancos de dados multidimensionais, um cubo é uma estrutura que organiza os dados em várias dimensões, permitindo análises e consultas OLAP (Online Analytical Processing). Um cubo contém informações numéricas chamadas fatos, que representam as medidas ou métricas que estão sendo analisadas. As dimensões, por outro lado, fornecem contextos para os fatos e representam categorias ou atributos pelos quais os dados podem ser agrupados e filtrados. As dimensões fornecem uma visão hierárquica dos dados e ajudam a navegar e analisar os fatos de diferentes perspectivas. Em resumo, os cubos, fatos e dimensões nos bancos de dados multidimensionais são componentes essenciais para facilitar a análise de dados multidimensionais e suportar consultas complexas e exploratórias.

6. Dado uma planilha de vendas de uma empresa do varejo, crie um modelo *Star Schema*.

	A	B	C	D	E	F	G
1	Tempo (Dia)	Produto (Item)	Geografia (Loja)	Valor da venda (R\$)	Quantidade de itens	Valor de venda (R\$)	...
2							
3	05/01/04	Lápis n? 2 – Faver Carel	Loja 04	78	65	1,2	...
4	05/01/04	Lápis n? 2 – Faver Carel	Loja 06	150	125	1,2	...
5	05/01/04	Caneta Clic azul - fina	Loja 04	117,6	84	1,4	...
6	05/01/04	Caneta Clic vermelha - fina	Loja 04	39,2	28	1,4	...
7
8	23/03/04	Caneta Clic azul - fina	Loja 06	123	82	1,5	...
9	23/03/04	Bloco recibo Jordel	Loja 12	132,5	53	2,5	...
10

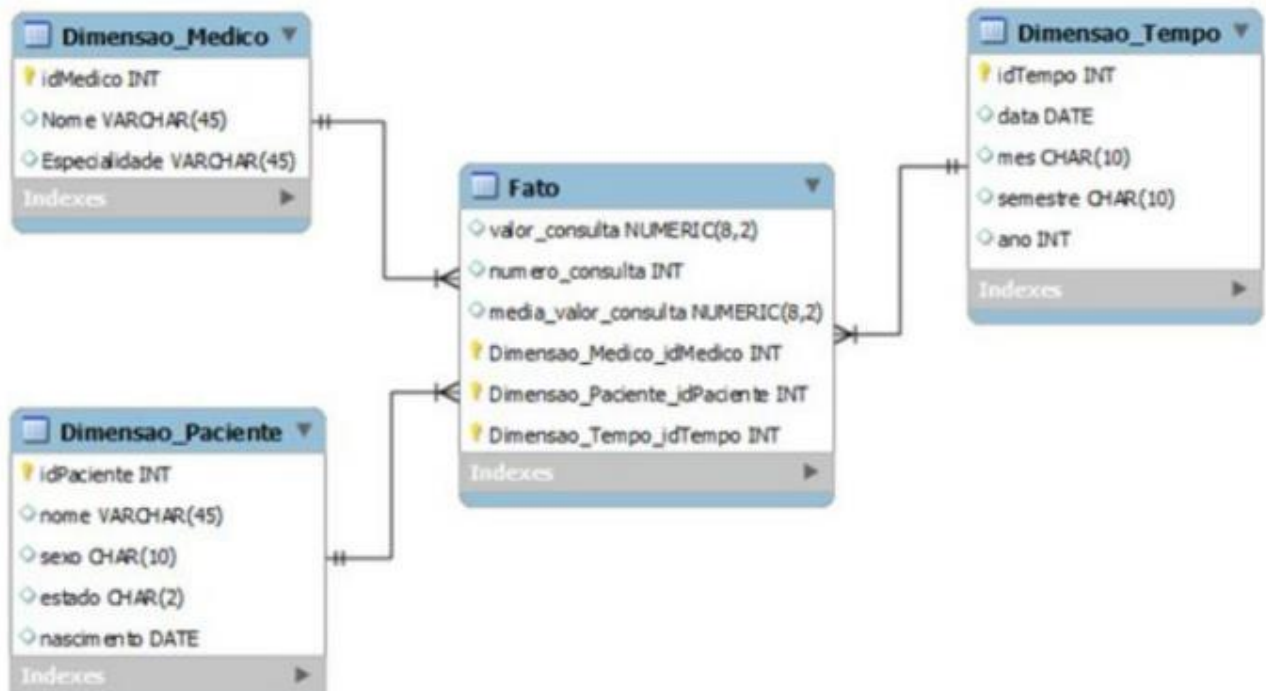
Figura 3 Planilha de Vendas.



7. Dado uma planilha de consultas diárias de uma clínica médica, crie um modelo *Star Schema*.

	A	B	C	D	E	F	G	H	I
1	Data	Valor_consulta	Médico	Especialidade	Paciente	Sexo	Data Nasc	Estado	...
2	05/01/11	125.50	Carla Beker	Ginecologia	Carla Rocha	Feminino	03/06/82	BA	...
3	23/03/11	111.67	Pedro Zanuncio	Dermatologia	Bruna Oliveira	Feminino	25/08/85	PR	...
4	03/07/11	124.47	Domingos	Pediatria	Caetano Queiroz da Silva	Masculino	12/10/10	MS	...
5	17/05/12	62.18	Ricardo Guirelli	Clinico Geral	Paulo Gomes	Masculino	10/04/80	SP	...
6
7	18/09/12	268.00	Andyane Tetila	Infectologia	Renato de Melo	Masculino	15/11/72	MS	...
8	08/10/13	141.18	Pedro Zanuncio	Dermatologia	Bruna Oliveira	Feminino	25/08/85	PR	...
9

Figura 4 Planilha de Consultas Diárias de uma Clínica Médica.



8. Quais as vantagens do Data Warehouse?

Entre as vantagens podemos destacar a consolidação de dados de várias fontes em um único local, análise de dados mais complexa e abrangente, o modelo dimensional para organização e navegação dos dados, a melhoria no desempenho das consultas analíticas e o suporte eficiente à tomada de decisões.

9. O que é um Data Mart?

Um Data Mart é um subconjunto de um Data Warehouse que contém dados específicos para atender a necessidades analíticas de um departamento, equipe ou área de negócio específica dentro de uma organização. Diferentemente do Data Warehouse, que abrange toda a empresa, o Data Mart é projetado para fornecer informações detalhadas e especializadas para usuários com necessidades específicas.

10. O que é um Data Lake? Como funciona um repositório de Data Lake Centralizado?

Um Data Lake é um repositório centralizado de dados brutos e diversos, sem a necessidade de estruturação prévia, permitindo armazenar grandes volumes de dados de diferentes fontes. Um repositório de Data Lake centralizado é um único local onde os dados são armazenados sem transformações, preservando sua integridade e possibilitando a aplicação de estruturação e organização dos dados no momento da análise.

11. Quais os fornecedores proeminentes de repositórios de Data Lake?

Amazon Web Services (AWS) com o serviço Amazon S3, Microsoft Azure com o serviço Azure Data Lake Storage e o serviço Azure Databricks e Google Cloud Platform (GCP) com o serviço Google Cloud Storage.

12. Quais os benefícios do Data Lake?

O Data Lake oferece benefícios como armazenamento flexível de dados brutos e não estruturados, escalabilidade para lidar com grandes volumes de dados, baixo custo de armazenamento, capacidade de explorar e descobrir insights avançados por meio de ferramentas

de big data, flexibilidade analítica para adaptar a estrutura dos dados e integração com ecossistemas de dados, permitindo uma análise abrangente e um melhor aproveitamento dos dados.

13. Quais as diferenças entre o *Data Lake* e *Data Warehouse*?

O Data Lake armazena dados brutos e não estruturados em sua forma original, sem a imposição de uma estrutura predefinida, oferecendo flexibilidade e capacidade de lidar com diversos tipos de dados. O Data Warehouse, por outro lado, segue uma abordagem estruturada, organizando os dados em um esquema predefinido, permitindo consultas analíticas pré-definidas e relatórios padronizados. Enquanto o Data Lake é mais flexível e ad-hoc, o Data Warehouse é otimizado para consultas rápidas e possui integração de dados mais rigorosa.