**1) Descreva com suas palavras os principais conceitos abaixo:**

**a) O que é um Data Warehouse?** Um data warehouse nada mais é do que um banco de dados feito principalmente para armazenar grandes volumes de dados e disponibiliza-los com muita velocidade para que times de negócio possam fazer analises e gerar insights. Geralmente os dados são coletados de sistemas transacionais, também chamados de OLTP, passam por um processo de tratamento e limpeza para enfim serem carregados para o data warehouse. Uma outra vantagem de se ter um DW é juntar dados de diferentes sistemas e armazena-los em um lugar só, por exemplo.: Uma empresa utiliza um sistema para registrar as vendas de seu ecommerce e outro para as vendas de sua loja física, tendo um data warehouse e um processo de ETL bem feito, todos os registros irão para um mesmo banco de dados, o que facilitará muito a criação de relatórios e análises.

**b) Quais características possuem as tabelas do tipo Fato e Dimensão?**

**Fato:** São tabelas onde ficarão armazenadas as medidas ou eventos de medição, também pode conter colunas de data do evento. Uma tabela fato terá todos os cálculos e valores a respeito do evento, para informações descritivas a tabela fato terá apenas colunas com o ID, que poderão ser utilizadas para fazer relacionamentos com a dimensão. São exemplos de fatos: venda, estoque, devolução etc.

**Dimensão:** São tabelas que armazenarão informações descritivas sobre os fatos, geralmente uma dimensão não terá colunas de valores e medições, apenas colunas com características dos fatos, por exemplo: informações sobre o produto, sobre uma localização ou um vendedor. Uma dimensão não poderá ter valores duplicados, cada registro deverá ter um ID único.

**c) O que é ETL?** É um processo de extração, tratamento e carga, geralmente os dados são coletados de um sistema transacional, também chamados e OLTP, passam por uma etapa de tratamento e limpeza, nesse processo os nomes das colunas podem ser alterados para que fiquem de uma forma mais padronizada, além disso é comum acontecer alguns filtros e cálculos. Por fim o dado será carregado em um banco de dados OLAP, onde ficará disponível para que sejam feitas analises, relatórios e dashboards de negócio.

**d) Quais são as principais atribuições de um Engenheiro de Dados?** O Engenheiro de Dados é o responsável pela criação de pipelines que transformam os dados brutos que estão nos mais variados formatos, desde bancos de dados transacionais até arquivos de texto, em um formato útil e fácil para a análise, também é responsabilidade de um engenheiro de dados garantir a qualidade e a segurança dessas informações para que empresas possam tomar decisões de acordo com esses dados.

**e) O que é Trade Marketing?** É uma área em quevendas e marketing atuam em conjunto para aumentar as vendas. Para isso, é necessário profissionais que dominam conceitos dessas duas áreas e que entendem as necessidades do consumidor. Um grande exemplo de trade marketing somado a análise de dados é de uma rede de supermercados norte-americana, que após algumas análises identificou que em véspera de finais de semana, um grande número de homens que compravam fraldas, também comprava cerveja consumo enquanto cuidavam das crianças e assistiam aos jogos na televisão durante o final de semana. A empresa somou a análise de dados com trade marketing e utilizou esse novo conhecimento para colocar lado a lado as gôndolas de fraldas e cervejas na rede de mercados, incrementando assim a venda conjunta dos dois produtos.

**2) Crie uma query, considerando o SGBD MySQL, para exibir todos os dados de uma tabela de Pontos de Venda (tabela origem PONTO\_VENDA\_UNIDADE) e restringir apenas os pontos de venda que possuem sell in maior que 20.000 (campo SELLIN) e ainda ordená-los por nome do ponto de venda (campo NOME\_PDV).**

SELECT

\*

FROM PONTO\_VENDA\_UNIDADE

WHERE SELLIN > 20000

ORDER BY NOME\_PDV;

**3) Considerando a tabela de origem da questão anterior, crie uma query que some o valor de sell in de acordo com cada ponto de venda e agrupe os resultados por mês (campo MES) e ano (campo ANO). Ordene os registros por um período cronológico de forma crescente e por nome do ponto de venda.**

SELECT

ANO,

MES,

NOME\_PDV,

SUM(SELLIN) AS SELLIN

FROM PONTO\_VENDA\_UNIDADE

GROUP BY ANO, MES, NOME\_PDV

ORDER BY ANO ASC, MES ASC, NOME\_PDV ASC;

**4) Considerando a tabela de origem da questão 2 e uma segunda tabela VISITAS\_PONTO\_VENDA, crie uma query que calcule a quantidade de visitas do ponto de venda de nome INVOLVES, sabendo-se que a tabela de visitas possui um campo que identifica se o ponto de venda foi visitado ou não chamado FL\_VISITADO (Se 1 = Ponto de venda visitado / Se 0 = Ponto de venda não visitado). O campo chave que liga as duas tabelas é ID\_PDV (na tabela PONTO\_VENDA\_UNIDADE) e FK\_PDV(na tabela VISITAS\_PONTO\_VENDA). A query deve mostrar apenas as informações de nome do ponto de venda e quantidade de visitas realizadas.**

SELECT

pdv.NOME\_PDV,

SUM(v.FL\_VISITADO) AS VISITAS

FROM VISITAS\_PONTO\_VENDA v

JOIN PONTO\_VENDA\_UNIDADE pdv ON pdv.ID\_PDV = v.FK\_PDV

WHERE pdv.NOME\_PDV = 'INVOLVES'

GROUP BY pdv.NOME\_PDV;

**5) Considerando a query abaixo, a pessoa engenheira de dados identificou que a performance da query está muito abaixo do esperado. Imaginando que um dos problemas possa estar relacionado aos índices das tabelas do banco de dados, a pessoa resolveu criar os índices nas tabelas. Liste quais possíveis campos devem ser indexados nas tabelas do banco de dados para que a query criada possa performar melhor. Leve em consideração que nenhum campo no banco de dados está indexado.**

**select**

**FT.CICLO,**

**FT.ID\_DIM\_PDV,**

**FT.ID\_BLOCO\_ITEM,**

**SUM(FT.QTD\_PONTO\_EXTRA),**

**SUM(FTPI.TOTAL\_NOTA\_ITEM)**

**from FT\_DOMINANCIA\_PONTO\_EXTRA\_COMPLIANCE FT**

**inner join TABREF\_PAINEL\_LOJAS\_LP TPLL**

**on FT.ID\_DIM\_PDV = TPLL.ID\_DIM\_PDV**

**and FT.CICLO = TPLL.CICLO**

**inner join FT\_PERFECTSTORE\_ITEM FTPI**

**on FT.CICLO = FTPI.CICLO**

**and FT.ID\_DIM\_PDV = FTPI.ID\_DIM\_PDV**

**and FT.ID\_BLOCO\_ITEM = FTPI.ID\_BLOCO\_ITEM**

**and FT.SEMANA\_LP = FTPI.SEMANA\_LP**

**where**

**FT.CICLO = 202009**

**and FT.ID\_DIM\_PDV = 223459792**

**group by FT.CICLO,**

**FT.ID\_DIM\_PDV;**

Nesse caso, minha sugestão seria incialmente criar índices para as colunas de ID, que são usadas nos inner join e no Where, ou seja, nas colunas abaixo:

- FT.ID\_DIM\_PDV

- TPLL.ID\_DIM\_PDV

- FTPI.ID\_DIM\_PDV

- FT.ID\_BLOCO\_ITEM

- FTPI.ID\_BLOCO\_ITEM

Se os problemas de performance continuarem existindo, seria importante criar índice para as demais colunas que são usadas nos dois inner join e no Where. Não recomendo criar índices para todas as colunas logo de cara pois isso pode impactar negativamente na inserção e atualização dos registros na tabela, visto que o banco de dados precisa manter os índices atualizados sempre que novos dados são inseridos ou modificados.

**6) Considere a instrução Python a seguir: x = [ print(i) for i in range(10) if i % 2 == 0 ] Após a execução dessa instrução no Python , a variável “x” conterá qual valor.**

O código basicamente irá imprimir os valores pares de 0 a 9 e não irá atribuir nenhum valor a variável X, portanto X será uma lista de valores None.

X = [None, None, None, None, None]

**7) Faça um script em Python que peça dois números e imprima a soma**

numero1 = float(input("Informe o primeiro número: "))

numero2 = float(input("Informe o segundo número: "))

soma = numero1 + numero2

print(f"A soma de {numero1} e {numero2} é {soma}")