

Teste Técnico: Pessoa Engenheira de Dados Involves

Olá candidato(a) ! Tudo bem ? Segue abaixo as questões relacionadas ao teste técnico de Engenharia de Dados da Involves.

Algumas dicas para você se dar bem no teste: Use suas palavras para responder as questões dissertativas; Tire um tempo para se concentrar e fazer o teste da melhor forma possível; Tente não deixar questões sem resposta mesmo que não saiba responder. Iremos avaliar também o esforço do candidato; E por fim, seja criativo, inove ! gostamos bastante dessa skill.

Estaremos na torcida para que você faça um bom teste e seja aprovado(a).

1) Descreva com suas palavras os principais conceitos abaixo:

a) O que é um Data Warehouse ?

R: Um DW é um repositório de dados históricos que podem armazenar dados estruturados e não estruturados de diversas fontes, tratados ou não (RAW Data), onde os dados podem ser explorados (data mining/data science) e também podem servir de base para construção da Data Lakes e utilizados em sistemas de B.I, que atendam à contextos específicos da área de negócio.

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

R: Uma tabela fato possui dados transacionais de uma empresa/contexto e necessita possuir ao menos uma métrica (campo de valor numérico, exemplo: quantidade de vendas, valor de produto, valor de faturamento, etc) e uma unidade de tempo (data onde ocorreu a transação), geralmente a tabela fato é o *core* de modelo estrela (apenas 1 fato e n dimensões) de um sistema B.I, quando o sistema de B.I possui mais de uma tabela fato, essa modelagem é conhecida como *Snowflake que possui 2 ou mais tabelas fatos vinculadas às dimensões*.

As tabelas do tipo dimensão, são tabelas descritivas, que irão determinar características inerentes à tabela fato, essas características são vinculadas às tabelas fatos através de chaves contidas em ambas tabelas.

c) O que é ETL ?

R: ETL é o acrônimo das etapas de “Extração, Transformação e Carga” (do inglês *extract, transform and load*). O ETL é uma das, senão a principal etapa em projetos de análise e tratamento de dados, pois no ETL são realizadas várias operações como limpeza/higienização dos dados, aplicação de lógica/regras de negócio, transformações e conversões de tipos de dados, aplicação de modelos de ML já consolidados.

Para a realização de um processo de ETL, podem ser utilizadas ferramentas visuais como o PDI (Pentaho Data Integration), SSIS (Sql Server Integration Services), AWS Glue Studio, Talend, entre outros. Como também linguagens de programação e script como Python, R, Scala e JavaScript.

Cada etapa do ETL é feita de forma separada e possui suas características:

- **Extração:** é a etapa onde é feita a coleta dos dados, de fontes diversas, podendo elas ser documentos (textos/arquivos), bancos de dados, web (scraping), streaming de redes sociais, etc. Os dados extraídos podem ser armazenados em repositórios temporários (chamados área de staging) ou DW's a depender da necessidade do negócio, onde serão consumidos no próximo processo de um trabalho de ETL.
- **Transformação:** São realizadas os tratamentos necessários para utilização em sistemas de B.I, análise exploratória dos Cientistas de Dados, criação de DW/Data Lakes, etc.
- **Carga:** Neste momento é realizada a entrega, dos dados que passaram pela etapa de transformação, essa entrega pode ser em diversas fontes como já comentado anteriormente (sistemas de B.I, relatórios, DataLakes, DW, entre outros).

d) Quais são as principais atribuições de um Engenheiro de Dados ?

R: De forma resumida seria realizar coleta, tratamento e disponibilização dos dados que serão consumidos na empresa (equipes de B.I, Cientistas de Dados, e usuários) e automatizar estes processos.

De forma mais detalhada, este trabalho pode ser feito utilizando a metodologia Crisp-DM (Cross Industry Standard Process for Data Mining) que consiste em 6 etapas:

1. Entendimento do negócio
2. Entendimento dos dados
3. Preparação dos dados
4. Modelagem/criação de modelos (ML) ou automação do processo em um pipeline de dados.
5. Validação dos Dados
6. Publicação dos dados

Onde as etapas 1 e 2 são realizadas até que sejam sanadas todas as dúvidas para seguir para as etapas 3 e 4, que são realizadas até que haja um produto que atenda aos requisitos levantados na etapa 1. Na etapa 5 esses requisitos são validados, caso necessite de correção, o processo retorna para a etapa 1 e reinicia o ciclo, caso contrário o processo criado estará pronto para publicação.

e) O que é Trade Marketing ?

R: Trade Marketing é uma operação *B2B (Business to Business)* que busca otimizar o processo de venda bem como melhorar os canais de comunicação dos distribuidores/fornecedores com as empresas que realizam a venda para o consumidor final

- 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 PV_UND
WHERE PV_UND.SELLIN >= 20000
ORDER BY PV_UND.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.

```
R1:
WITH PV_UND AS (
    SELECT
    *
    FROM PONTO_VENDA_UNIDADE PV_UND
    WHERE PV_UND.SELLIN >= 20000
    ORDER BY PV_UND.NOME_PDV
)

SELECT
    ANO,
    MES,
    PERFIL_PDV,
    TIPO_PDV,
    ID_PDV,
    NOME_PDV,
    SUM(SELLIN) AS SELLIN_TOTAL
FROM PV_UND
GROUP BY
    ANO,
    MES,
    PERFIL_PDV,
    TIPO_PDV,
    ID_PDV,
    NOME_PDV
ORDER BY ANO,MES,NOME_PDV;
```

R2:

```
SELECT
    T.ANO,
    T.MES,
    T.PERFIL_PDV,
    T.TIPO_PDV,
    T.ID_PDV,
    T.NOME_PDV,
    SUM(T.SELLIN) AS SELLIN_TOTAL
FROM (
    SELECT
        *
        FROM PONTO_VENDA_UNIDADE PV_UND
        WHERE PV_UND.SELLIN >= 20000
        ORDER BY PV_UND.NOME_PDV
    ) T
GROUP BY
    ANO,
    MES,
    PERFIL_PDV,
    TIPO_PDV,
    ID_PDV,
    NOME_PDV
ORDER BY ANO,MES,NOME_PDV;
```

- 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.

```
WITH PV_UND AS (  
    SELECT  
        *  
    FROM PONTO_VENDA_UNIDADE PV_UND  
  
    WHERE PV_UND.SELLIN >= 20000  
    ORDER BY PV_UND.NOME_PDV  
)  
  
SELECT  
    PV_UND. NOME_PDV,  
    COUNT(VPV.*) AS QTD_VISITAS  
FROM PV_UND  
  
INNER JOIN VISITAS_PONTO_VENDA VPV  
    ON VPV.FK_PDV = PV_UND.ID_PDV  
  
WHERE UPPER(PV_UND. NOME_PDV) = 'INVOLVES'  
    AND VPV.FL_VISITADO = 1  
  
GROUP BY PV_UND. 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;
```

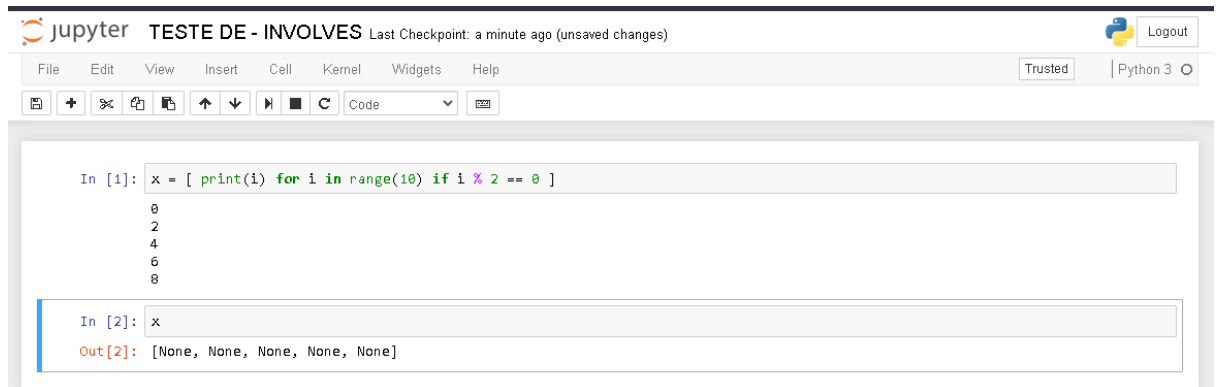
R: Os campos ID_DIM_PDV, CICLO devem ser indexados na tabela TABREF_PAINEL_LOJAS_LP

Os campos ID_DIM_PDV, CICLO, ID_BLOCO_ITEM, SEMANA_LP devem ser indexados nas tabelas: FT_DOMINANCIA_PONTO_EXTRA_COMPLIANCE e FT_PERFECTSTORE_ITEM

OBS.: Caso o campo “CICLO” seja armazenado como VARCHAR() na cláusula WHERE a consulta do CICLO entre aspas ‘202009’ tende a performar melhor também.

- 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.

R: `[None, None, None, None, None]`



Jupyter interface showing the execution of the following code:

```
In [1]: x = [ print(i) for i in range(10) if i % 2 == 0 ]
```

Output of the first cell:

```
0
2
4
6
8
```

Second cell execution:

```
In [2]: x
```

Output of the second cell:

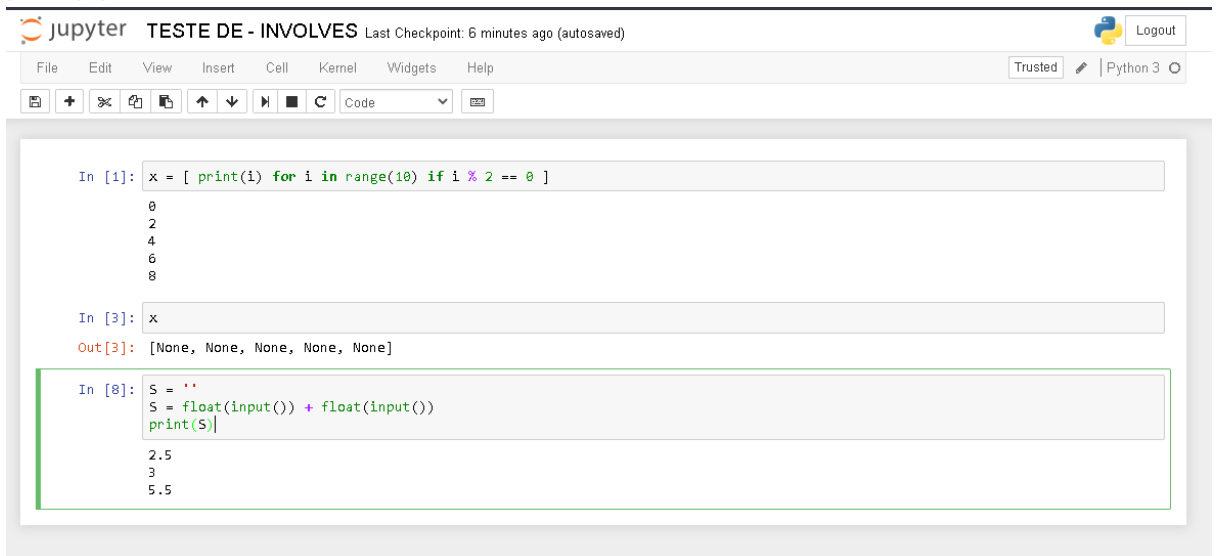
```
Out[2]: [None, None, None, None, None]
```

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

S = "

`S = float(input()) + float(input())`

`print(S)`



Jupyter interface showing the execution of the following code:

```
In [1]: x = [ print(i) for i in range(10) if i % 2 == 0 ]
```

Output of the first cell:

```
0
2
4
6
8
```

Second cell execution:

```
In [3]: x
```

Output of the second cell:

```
Out[3]: [None, None, None, None, None]
```

Third cell execution:

```
In [8]: S = ''
S = float(input()) + float(input())
print(S)
```

Output of the third cell:

```
2.5
3
5.5
```

- 8) Para responder às questões 8, 9 e 10 utilize a ferramenta Pentaho Data Integration (PDI) na versão de sua preferência. A ETL final deve conter um job principal que, por sua vez, deve conter as transformações criadas nas questões 8, 9, 10. Além disso, que tal ganhar um ponto a mais nessas questões ? Para isso, inclua o projeto criado em um repositório do Github (é importante que seja público para termos visibilidade, ok ?). Compartilhe por aqui o link para o repositório.

R: <https://github.com/EsliAraujo/Involves>

Segue as questões:

Construa uma transformação que deve usar como datasource o dataset (DATASET_TESTE_DE.csv) que contém informações de coletas de dados nos ponto de vendas. A ETL deve consultar o dataset e inserir, em uma base de dados (modelo dimensional), as informações coletadas, conforme as tabelas abaixo:

- a) Dimensão Calendário (DIM_CALENDARIO): Deve conter data, mês e ano da coleta
 - b) Dimensão Ponto de Venda (DIM_PDV): Deve conter o id, nome e perfil do ponto de venda
 - c) Dimensão Linha de Produto (DIM_LINHA_PRODUTO): Deve conter o id, nome e perfil da linha de produto
- 9) Construa uma transformação que deve usar como datasource o dataset (DATASET_TESTE_DE.csv) que contém informações de coletas de dados nos ponto de vendas. A transformação deve consultar o dataset e inserir, em uma base de dados (modelo dimensional), as informações coletadas, conforme as tabelas abaixo:
- a) Fato Disponibilidade (FT_DISPONIBILIDADE): Deve conter os ids de ligação das tabelas de dimensões criadas na questão anterior e a quantidade de presenças de cada linha de produto no mês de Setembro/20.
 - b) Fato Disponibilidade Agregada (FT_DISPONIBILIDADE_AGREGADA): Deve conter os ids de ligação das tabelas de dimensões (Dimensão Calendário e Ponto de Venda) e a quantidade de presença de linhas de produto agrupadas por ponto de venda no mês de Setembro/20.

Obs: Os dados de “Disponibilidade” estão categorizados na coluna TIPO_COLETA com o valor “Disponibilidade”. A presença é contada sempre que no campo VALOR aparecer o valor “SIM”

- 10) Construa uma transformação que deve usar como datasource o dataset (DATASET_TESTE_DE.csv) que contém informações de coletas de dados nos ponto de vendas. A transformação deve consultar o dataset e inserir, em uma base de dados (modelo dimensional), as informações coletadas, conforme as tabelas abaixo:

- a) **Fato Ponto Extra (FT_PONTO_EXTRA):** Deve conter os ids de ligação das tabelas de dimensões criadas na questão anterior e a soma de ponto extras de cada linha de produto no mês de Setembro/20.
- b) **Fato Ponto Extra Agregada (FT_PONTO_EXTRA_AGREGADA):** Deve conter os ids de ligação das tabelas de dimensões (Dimensão Calendário e Ponto de Venda) e a soma de ponto extras de linhas de produto agrupadas por ponto de venda no mês de Setembro/20.

Obs: Os dados de “Ponto Extra” estão categorizados na coluna TIPO_COLETA com o valor “Ponto Extra”.