**Questão 1:** Considere a seguinte estrutura de tabelas:

|  |  |  |
| --- | --- | --- |
| **Pacientes** | **Atendimento** | **Prontuário** |
| id\_paciente | id\_atendimento | id\_prontuario |
| nome\_paciente | id\_paciente | id\_paciente |
| tel\_paciente | data\_atendimento | id\_atendimento |
| email\_paciente | hr\_atendimento | descricao\_prontuario |
| status\_paciente | nome\_unidade |  |
| data\_nascimento | endereco\_unidade |  |
|  | cd\_especialidade |  |

1. Escreva uma consulta SQL que retorne todos os atendimentos do paciente com a especialidade "Neurologia" nos últimos 6 meses.

SELECT a.id\_atendimento, p.nome\_paciente, a.cd\_especialidade

FROM atendimento a

INNER JOIN pacientes p ON a.id\_paciente = p.id\_paciente

WHERE a.cd\_especialidade = 'Neurologia' and data\_atendimento >= current\_date - interval '6 months';

1. Suponha que você precise obter todos os prontuários associados ao paciente de ID 123, incluindo os detalhes do atendimento relacionado a cada prontuário. A consulta deve retornar o nome do paciente, a descrição do prontuário e a data do atendimento correspondente.

SELECT pct.nome\_paciente, p.descricao\_prontuario, a.data\_atendimento

FROM prontuario p

INNER JOIN pacientes pct ON p.id\_paciente = pct.id\_paciente

INNER JOIN atendimento a ON a.id\_paciente = pct.id\_paciente

WHERE pct.id\_paciente = 123

**Questão 2:** Um cliente relata que está enfrentando lentidão em um determinado recurso do sistema. Como você abordaria esse problema para identificar a causa raiz e propor uma solução?

Identificação do problema:

A minha primeira abordagem seria buscar entender qual recurso está dando problema, solicitando ao usuário onde e quando começou;

Buscaria isolar o problema tentando entender se está relacionado a rede ou recursos físicos do sistema do usuário;

Verificaria os status e logs em busca de erros ou falhas;

Verificaria se existem chamados com descrições parecidas.

Ao identificar a causa raiz:

Se o problema fosse com máquina do usuário proporia melhorias, como aumento de RAM, Armazenamento, redes ou até mesmo upgrade de dispositivo.

Se o problema for o nosso sistema como falta de recursos solicitaria aumento ou melhorias de recursos

Se o problema for exclusivamente no recurso relatado pelo usuário buscaria verificar se há algum gargalho no código ou solicitaria ao responsável uma possível refatoração.

**Questão 3:** Um usuário informa que está recebendo uma mensagem de erro ao tentar realizar uma operação específica no sistema. Explique como você investigaria esse problema e quais etapas seguiria para solucioná-lo.

Identificação do problema:

Solicitaria a mensagem e o código do erro;

Pediria ao usuário descreva como reproduzir o erro novamente;

Analisaria se há outros chamados relacionado ao mesmo problema.

Etapas a serem seguidas para solucionar:

Se o erro estiver relacionado com possíveis falhas nos códigos direcionaria ao responsável ou tentaria sanar o problema;

Se o problema estiver relacionado ao recurso indisponível orientaria ao usuário que deverá aguardar

Se o erro estiver relacionado com os inputs do usuário orientaria a realizar as ações novamente com cuidado preenchendo ou enviado aquilo que se pede

**Questão 4:** Descreva como você lidaria com uma situação em que um usuário encontra um erro desconhecido.

Eu lidaria solicitando ao usuário que descreva os passos seguidos para reproduzir o erro

Perguntaria quando o problema começou

Proporia revisão nos códigos, novas validações ou atualizações.

**Questões de Lógica:** Usando uma das seguintes linguagens de programação: C, C++, Java, C#, Javascript ou Python, resolva as questões a seguir.

**Questão 1:** Faça um programa, em seguida, um programa que solicite a entrada de dois números, em seguida imprime na tela o quadrado do menor número e a raiz quadrada do maior número, se for possível.

from math import sqrt

def mostrar\_raiz(num):

if(num >= 0):

print("Raiz: " + str(sqrt(num)))

def mostrar\_quadrado(num):

print("Quadrado: " + str(num \* num))

try:

num1 = float(input("Digite um numero: "))

num2 = float(input("Digite outro numero: "))

if(num1 <= num2):

mostrar\_raiz(num1)

mostrar\_quadrado(num2)

else:

mostrar\_raiz(num2)

mostrar\_quadrado(num1)

except ValueError:

print("O valor de entrada deve ser um numero")

except:

print("Erro desconhecido")

**Questão 2:** Faça um programa que solicite ao usuário o ano em que ele nasceu e verifique se o ano é Bissexto ou não e imprima uma mensagem na tela.

try:

year = int(input("Digite um ano: "))

if(year % 4 == 0 and year % 100 != 0 or year % 400 == 0):

print("O ano {} é bissexto".format(year))

else:

print("O ano {} não é bissexto".format(year))

except ValueError:

print("O valor de entrada deve ser um numero")

except:

print("Erro desconhecido")