

Instruções

Atividade Avaliativa de Banco de dados II Referente as aulas de: - Junções - Subconsultas - CTE - Funções de Janela - Indexação - Planos de acesso A prova estará disponível a partir das 11 horas do dia 28/08/2024 e se encerrara as 18:00 horas de 04/09/2024. A avaliação é individual, online e com consulta. Boa prova!

Questão:1

Marque as alternativas que são VERDADEIRAS para as afirmações abaixo referente à junções de tabelas em bancos de dados:

- a) JOIN é uma cláusula utilizada para combinar dados de duas ou mais tabelas em uma única consulta.
- b) O INNER JOIN entre duas tabelas retorna apenas as linhas que possuem correspondência em ambas as tabelas.
- c) O LEFT JOIN entre duas tabelas retorna todas as linhas da tabela da esquerda e todas as linhas da tabela da direita.
- d) O RIGHT JOIN entre duas tabelas retorna todas as linhas da tabela da direita e apenas as linhas correspondentes da tabela da esquerda.
- e) O FULL JOIN entre duas tabelas retorna todas as linhas de ambas as tabelas gerando um resultado derivado de um plano cartesiano entre os dados das tabelas.
- f) O CROSS JOIN entre duas tabelas combina cada linha da tabela da esquerda com todas as linhas da tabela da direita.

Peso: 10,00% Tipo de pergunta: Múltipla escolha

Questão:2

Marque VERDADEIRO ou FALSO para as afirmações abaixo e depois escolha a alternativa que corresponde a sequência correta referente a índices em bancos de dados?

- () Um índice nonclustered é criado na(s) coluna(s) que define a chave primária da tabela.
- () Um índice simples é criado em várias colunas.
- () Um índice composto pode ter no mínimo uma coluna.
- () Um índice de cobertura (criado com INCLUDE) é um índice que não cobre uma lista de select resolvendo problemas de lookup, mas é criado apenas para melhorar o desempenho da atualização de registros.
- () Um índice nonclustered é criado em uma coluna ou conjunto de colunas que não são frequentemente usadas para consultas.
- () Um índice clustered é criado em uma coluna ou conjunto de colunas que não são a chave primária, mas são frequentemente usadas para consultas.

- a) F, F, F, F, F, F
- b) V, F, F, F, F, F
- c) V, F, V, F, F, F
- d) V, F, V, V, F, F
- e) F, F, F, V, F, F

Peso: 10,00% Tipo de pergunta: Objetiva

Questão:3

Dada a consulta abaixo, o que é possível fazer para tornar a consulta mais otimizada possível (considere apenas a melhor resposta - mais completa):

```
SELECT NO_MUNICIPIO_ESC  
FROM MICRODADOS_ENEM_2022_SC  
WHERE NO_MUNICIPIO_ESC = 'Criciúma'
```

Para responder essa pergunta, considere o seguinte cenário:

A tabela MICRODADOS_ENEM_2022_SC possui índice clustered pk_MICRODADOS_ENEM_2022_SC criado com a chave NU_INSCRICAO.

A tabela MICRODADOS_ENEM_2022_SC possui índice nonclustered idx1 criado com a chave NO_MUNICIPIO_ESC.

- a) Criar um novo índice nonclustered chamado idx2 com a cláusula include na coluna NO_MUNICIPIO_ESC.
- b) Dropar o índice nonclustered idx1 e deixar somente o índice clustered.
- c) A consulta já está a mais otimizada possível.
- d) Criar um novo índice clustered chamado pk_MICRODADOS_ENEM_2022_SC_2 com a chave NO_MUNICIPIO_ESC para ser mais rápido.
- e) Dropar o índice clustered pk_MICRODADOS_ENEM_2022_SC e recriá-lo com a chave NO_MUNICIPIO_ESC.

Peso: 10,00% Tipo de pergunta: Objetiva **Questão:4**

Dada a consulta abaixo, o que é possível fazer para eliminar o problema de Key Lookup (considere todas as alternativas possíveis):

```
SELECT NO_MUNICIPIO_ESC, TP_ESCOLA  
FROM MICRODADOS_ENEM_2022_SC  
WHERE NO_MUNICIPIO_ESC = 'Treviso'
```

Para responder essa pergunta, considere o seguinte cenário:

A tabela MICRODADOS_ENEM_2022_SC possui índice clustered pk_MICRODADOS_ENEM_2022_SC criado com a chave NU_INSCRICAO.

A tabela MICRODADOS_ENEM_2022_SC possui índice nonclustered idx1 criado com a chave NO_MUNICIPIO_ESC.

- a) Criar um novo índice nonclustered chamado idx2 com a chave NO_MUNICIPIO_ESC e com a cláusula include na coluna TP_ESCOLA.
- b) Dropar o índice nonclustered idx1 e recriá-lo com a chave NO_MUNICIPIO_ESC e com a cláusula include na coluna TP_ESCOLA.
- c) Não existe problema de Key Lookup na consulta.
- d) Criar um novo índice clustered chamado pk_MICRODADOS_ENEM_2022_SC_2 com a cláusula INCLUDE na coluna TP_ESCOLA.
- e) Dropar o índice clustered pk_MICRODADOS_ENEM_2022_SC e recriá-lo com a cláusula INCLUDE na coluna TP_ESCOLA.

Peso: 10,00% Tipo de pergunta: Múltipla escolha **Questão:5**

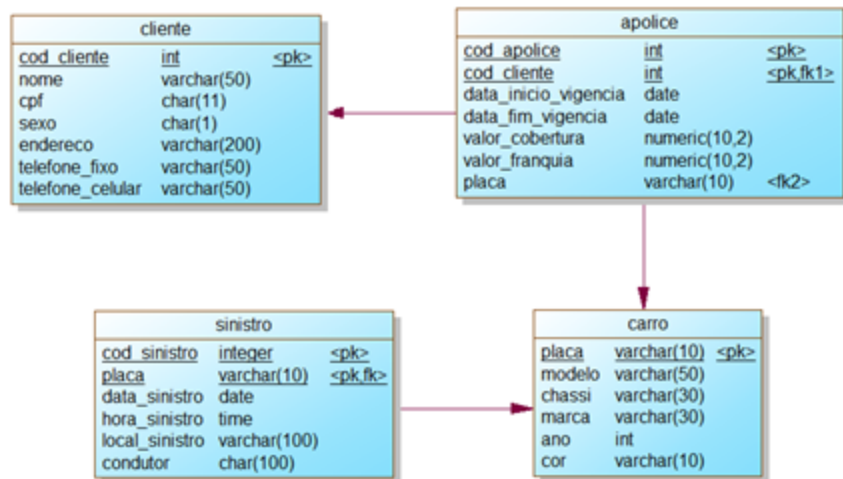
Construa uma consulta que informe quais carros possuem apólice não vigente, ou seja, apólice vencida. Considere mostrar todos os dados dos carros (todas as colunas da tabela carro) na sua consulta.

NOTA 1: A sua consulta deverá ser construída através de **subconsultas (alinhada ou correlacionada ou ambas)**. JOIN não será aceito como resposta válida.

NOTA 2: A resposta desta questão deverá ser o código completo da consulta.

NOTA 3: Considere apólices não vigentes como aquelas que tem a data de fim da vigência inferior a data atual.

NOTA 4: Você pode utilizar a função getdate() para ter acesso a data atual.

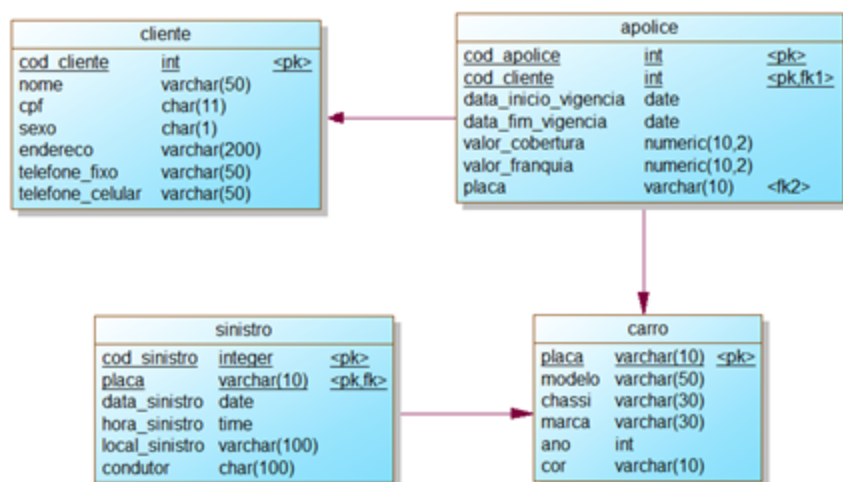


Peso: 10,00% Tipo de pergunta: Dissertativa **Questão:6**

Construa uma **consulta** que liste todos os dados de sinistro, adicionando uma nova coluna chamada **nr_seq** ao final da lista de select, (não é para criar uma coluna com ALTER TABLE, usar somente SELECT), com um contador incremental de sinistros ordenado de forma ascendente por data e hora do sinistro.

NOTA 1: A resposta desta questão deverá ser o código completo da consulta.

NOTA 2: Utilize função de janela para resolver essa questão.



O resultado da consulta deve seguir este exemplo (não necessariamente os mesmos dados, mas sim a disposição e classificação das colunas):

cod_sinistro	placa	data_sinistro	hora_sinistro	local_sinistro	condutor	nr_seq
202255507	MZT1826	2022-05-08	18:00:00.0000000	NOVA VENEZA	LUCAS	1
202255503	IAC8974	2022-05-24	22:40:00.0000000	TUBARAO	CUNHA	2
202255509	IAC8974	2022-08-08	07:00:00.0000000	CRICIUMA	TEO	3
202255508	IAC.8974	2022-08-08	21:00:00.0000000	CRICIUMA	PEDRO	4

Peso: 10,00% Tipo de pergunta: Dissertativa **Questão:7**

1a parte

Crie uma consulta que liste o nome de todos os estados e a quantidade de municípios vinculados a estes estados. Utilizar JOIN.

2a parte

Após isso, adicione na sua consulta, uma coluna de ranking de classificação, ordenador pela quantidade de municípios do maior para o menor (descendente), sem pular a sequência da contagem considerando, para os casos de empate o mesmo

número de posição.

A numeração do ranking deve respeitar um sequencial número (sem saltos).

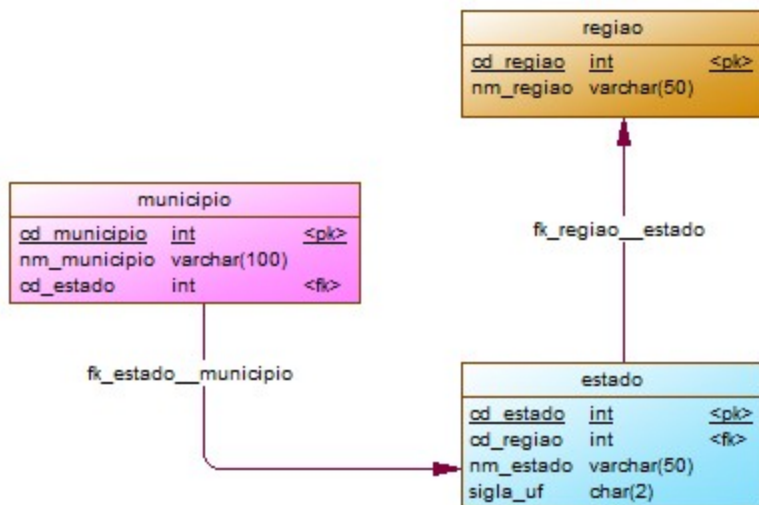
O resultado desta consulta deverá ser ordenado de forma descendente pela coluna de quantidade de municípios vinculados a cada estado.

REGRA IMPORTANTE: A 1a parte da consulta deverá estar dentro de uma CTE e a 2a parte deve ser um SELECT na CTE que você criou.

A resposta deverá ser o comando completo (parte 1 + parte 2).

Caso a resposta não esteja dentro da regra acima citada, não será considerada correta.

IMPORTANTE: O resultado final de sua consulta deverá ter apenas 3 colunas na *lista de select* (nome do estado, quantidade de município e ranking).

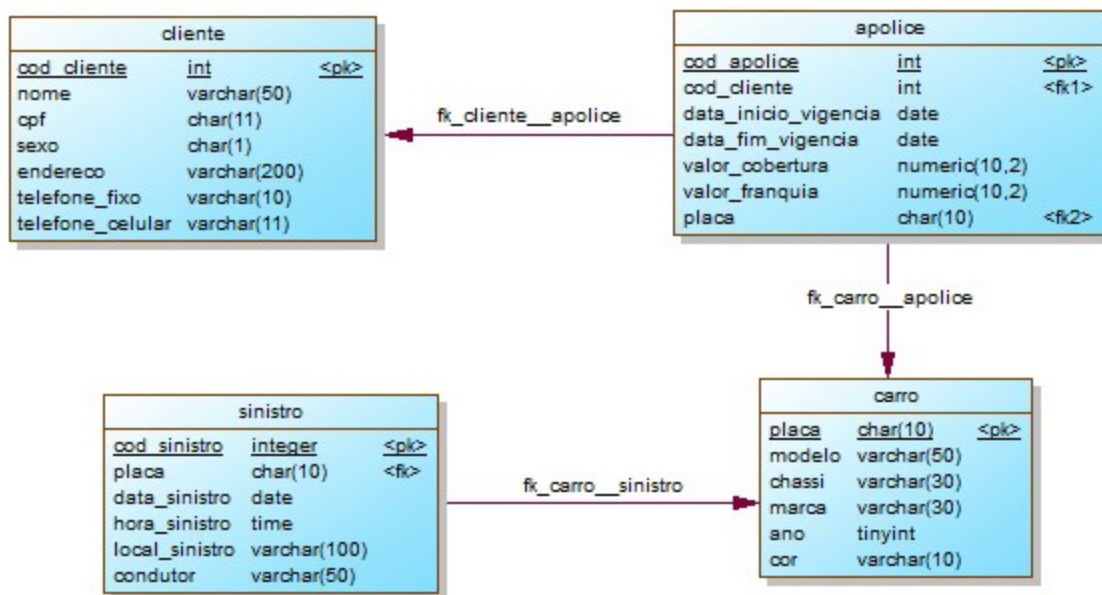


Peso: 10,00% Tipo de pergunta: Dissertativa **Questão:8**

Faça uma consulta na tabela apolice (listar todas as colunas), trocando a coluna cod_cliente pelo nome do cliente (usar uma subconsulta no *select list* para isso).

Depois, adicione uma nova coluna ao final do select list com o valor acumulado da coluna valor_franquia, particionado por nome do cliente (ASC), e ordenado pelo nome do cliente (ASC) e pelo código da apólice (ASC) respectivamente.

DICA: Utilize CTE para poder usar o nome do cliente dentro da função de janela.



Questão:9

Crie uma CHAVE PRIMARIA para a tabela MICRODADOS_ENEM_2022_SC chamado pk_MICRODADOS_ENEM_2022_SC usando a coluna NU_INSCRICAO como chave.

Resolva todos os impedimentos que você encontrar para criação da chave na tabela e coluna citados.

A sua resposta deverá conter todos os comandos utilizados, em ordem cronológica para criação da PK.

DICA: Utilize o comando ALTER TABLE.

Questão:10

1a parte

Crie um índice nonclustered chamado "idx_placa" na tabela "apolice", usando como chave a coluna que contém os dados das placas dos carros.

Este índice deverá cobrir (cobertura) as colunas Código da Apólice, Data de Início da Vigência e Data de Fim da Vigência, com o objetivo de evitar problemas com Key Lookup e/ou RID Lookup.

Sua resposta deverá ser o comando completo de criação do índice.

2a parte

Execute a consulta a seguir e informe o **nome do tipo de operador de plano de acesso** e o **nome do índice** que ele usou para execução em cada uma das etapas abaixo:

Etapla 1: Antes da criação do índice feita na 1a parte.

Etapla 2: Depois da criação do índice feita na 1a parte.

```
SELECT cod_apolice, data_inicio_vigencia, data_fim_vigencia
FROM apolice
WHERE placa = 'FFR1234'
```

apolice		
cod_apolice	int	<pk>
cod_cliente	int	<fk1>
data_inicio_vigencia	date	
data_fim_vigencia	date	
valor_cobertura	numeric(10,2)	
valor_franquia	numeric(10,2)	
placa	char(10)	<fk2>