EXERCÍCIO DE FIXAÇÃO + REVISÃO PROVA

1. SELECT CURSO.NM\_CURSO, COUNT(ALUNO.CD\_ALUNO)

FROM CURSO

LEFT JOIN ALUNO

ON CURSO.CD\_CURSO = ALUNO.CD\_CURSO

GROUP BY CURSO.NM\_CURSO

ORDER BY COUNT(ALUNO.CD\_ALUNO) DESC

1. SELECT \*

FROM CARRO

WHERE PLACA IN ( SELECT PLACA

FROM APOLICE

WHERE DATA\_FIM\_VIGENCIA >= GETDATE())

1. SELECT ESTADO.NM\_ESTADO, COUNT(MUNICIPIO,CD\_MUNICIPIO) AS QTDE

FROM ESTADO

INNER JOIN MUNICIPIO

ON ESTADO.CD\_ESTADO = MUNICIPIO.CD\_MUNICIPIO

GROUP BY ESTADO.NM\_ESTADO

HAVING COUNT(MUNICIPIO,CD\_MUNICIPIO) > 200

ORDER BY COUNT(MUNICIPIO,CD\_MUNICIPIO) DESC

1. SELECT \*, ROW\_NUMBER() OVER(ORDER BY DATA\_SNISTRO,HORA\_SNISTRO ASC) AS NR\_SEQ

FROM SINISTRO

REVISÃO PROVA

1. WITH CTE AS (

SELECT MODALIDADE, NOME\_ATLETA, TEMPO, DENSE\_RANK() OVER(PARTITION BY MODALIDADE ORDER BY TEMPO ASC) AS RANK

FROM RESULTADO\_NATACAO

WHERE DATA\_COMPETICAO = ‘2023-09-17')

SELECT \*

FROM CTE

WHERE RANK <= 3

1. TRANSAÇÕES SÃO OPERAÇÕES QUE MODIFICAM OS DADOS SELECINADOS, ALTERANDO NOME, INCLUINDO OUTROS DADOS OU AFETANDO VALORES. É IMPORTANTE MANTER O CONTROLE DE CONCORRÊNCIA PARA OUTRA PESSOA NÃO CONSIGUIR LER OU ALTERAR O DADO QUE ESTÁ EM ESTADO TRANSITÓRIO.
2. V, V, F, F, V
3. ELE MELHORA A PERFORMANCE DE CONSULTAS POIS PERMITE QUE OS DADOS SEJAM ORGANIZADOS E ‘PROCURADOS’ DE MANEIRA MAIS DIRETA. JÁ EM BLOQUEIOS DE CONCORRÊNCIA, que as transações acessem apenas as partes relevantes dos dados solicitados.
4. V, V, V, V, V
5. F, F, V, F, V, F
6. CREATE NONCLUSTERES INDEX ID1 ON RESULTADO\_NATACAO (DATA\_COMPETICAO) INCLUDE (MODALIDADE, NOME\_ATLETA, TEMPO);
7. SELECT AGENDA.COD\_AGENDA, PACIENTE.NM\_PACIENTE, PROFISSIONAL.NM\_PREOFISSIONAL, AGENDA.DT\_INICIO, AGENDA.DT\_FIM

FROM PACIENTE

INNER JOIN AGENDA

ON PACIENTE.CD\_PACIENTE = AGENDA.CD\_PACIENTE

INNER JOIN PROFISSIONAL

ON AGENDA.CD\_PROFISSIONAL = PROFISSIONAL.CD\_PROFISSIONAL

WHERE AGENDA.DT\_INICIO BETWEEN (‘2020-01-01' AND ‘2020-12-31') AND AGENDA.DT\_FIM BETWEEN (‘2020-01-01' AND ‘2020-12-31')

ORDER BY AGENDA.DT\_INICIO ASC

1. SELECT NM\_PROFISSIONAL

FROM PROFISSIONAL

WHERE *NOT EXISTS* (SELECT CD\_PROFISSIONAL

FROM AGENDA

WHERE (AGENDA.CD\_PROFISSIONAL = PROFISSIONAL.CD\_PROFISSIONAL) AND DT\_INICIO BETWEEN ‘2021-01-01' AND ‘2021-12-31')

1. SELECT PROFISSIONAL.NM\_PROFISSIONAL

FROM PROFISSIONAL

LEFT JOIN AGENDA

ON PROFISSIONAL.CD\_PROFISSIONAL = AGENDA.CD\_PROFISSIONAL

WHERE AGENDA.CD\_AGENDA IS NULL