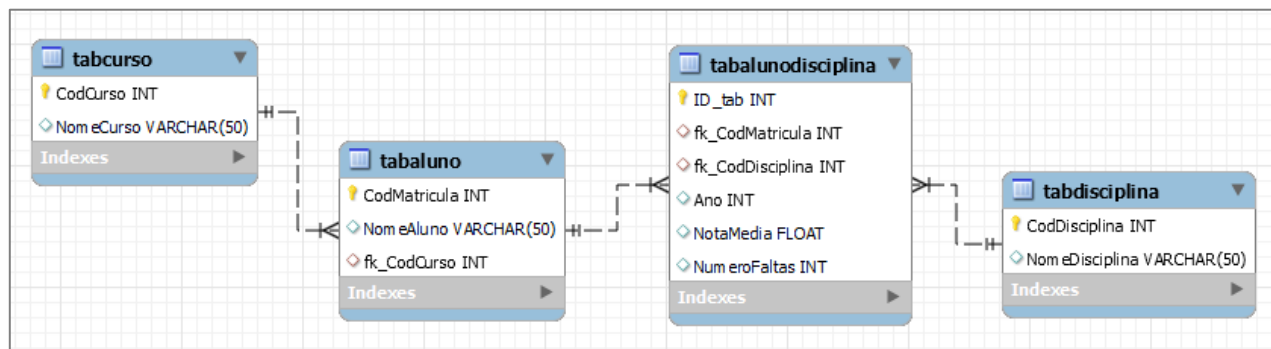


Nome: _____ Curso: _____ Data: _____

QUESTÃO 1) (8,0 pontos) Considere o seguinte esquema (Projeto Lógico) de um Banco de Dados:

(na Prova 2 Somativa, serão 3 sub-questões)



Considere que o esquema apresentado acima possui a seguinte instância:

ID_tab	fk_CodMatricula	fk_CodDisciplina	Ano	NotaMedia	NumeroFaltas
1	1	2	2019	10	10
2	2	1	2019	10	20
3	3	3	2020	10	10
4	1	4	2020	5	20
5	2	5	2020	10	10
6	3	2	2021	5	20

CodMatricula	NomeAluno	fk_CodCurso
1	Maria Flores	1
2	José Silva	2
3	Arthur Fontes	3
4	Fernando Santos	1
5	Carlos Torres	2
6	Antônia Mendes	3

CodDisciplina	NomeDisciplina
1	Matemática
2	Língua Portuguesa
3	Biologia
4	Lógica de Programação
5	Circuitos Integrados
6	Capacitores
7	Banco de Dados

CodCurso	NomeCurso
1	Técnico em Informática
2	Técnico em Eletrônica
3	Técnico em Administração

Apresente o **resultado obtido** para cada uma das consultas a seguir, no formato de tabela, e **explique** o que o comando faz:

***** Três questões sobre Consulta + Explicação sobre o que o comando faz *****

a)

```
SELECT      TC.NomeCurso, TA.NomeAluno
FROM        tabaluno AS TA
INNER JOIN  tabcurso AS TC ON (TA.fk_CodCurso = TC.CodCurso)
ORDER BY   TC.NomeCurso, TA.NomeAluno;
```

b)

```
SELECT      TC.NomeCurso, COUNT(*) AS TotalAlunos
FROM        tabaluno AS TA
INNER JOIN  tabcurso AS TC ON (TA.fk_CodCurso = TC.CodCurso)
GROUP BY    TC.NomeCurso
ORDER BY    TC.NomeCurso;
```

c)

```
SELECT      TC.NomeCurso, SUM(TAD.NumeroFaltas) AS TotalFaltasAlunos
FROM        tabaluno AS TA
INNER JOIN  tabcurso AS TC ON (TA.fk_CodCurso = TC.CodCurso)
INNER JOIN  tabAlunoDisciplina AS TAD ON (TAD.fk_CodMatricula = TA.CodMatricula)
GROUP BY    TC.NomeCurso
ORDER BY    TC.NomeCurso;
```

d)

```
SELECT      TC.NomeCurso, SUM(TAD.NumeroFaltas) AS TotalFaltasAlunos2020
FROM        tabaluno AS TA
INNER JOIN  tabcurso AS TC ON (TA.fk_CodCurso = TC.CodCurso)
INNER JOIN  tabAlunoDisciplina AS TAD ON (TAD.fk_CodMatricula = TA.CodMatricula)
WHERE       TAD.Ano = 2020
GROUP BY    TC.NomeCurso
ORDER BY    TC.NomeCurso;
```

e)

```
SELECT      TD.NomeDisciplina AS Disciplina, COUNT(*) AS TotalAlunos
FROM        tabDisciplina AS TD
INNER JOIN  tabAlunoDisciplina AS TAD ON (TD.CodDisciplina = TAD.fk_CodDisciplina)
GROUP BY    TD.NomeDisciplina
ORDER BY    TD.NomeDisciplina;
```

f)

```
SELECT      TD.NomeDisciplina AS Disciplina, COUNT(*) AS TotalAlunos,
SUM(NumeroFaltas) AS TotalFaltas
FROM        tabDisciplina AS TD
INNER JOIN  tabAlunoDisciplina AS TAD ON (TD.CodDisciplina = TAD.fk_CodDisciplina)
GROUP BY    TD.NomeDisciplina
ORDER BY    TD.NomeDisciplina;
```

g)

```
SELECT      TD.NomeDisciplina AS Disciplina, COUNT(*) AS TotalAlunos,
AVG(TAD.NotaMedia) AS NotaMédia
FROM        tabDisciplina AS TD
INNER JOIN  tabAlunoDisciplina AS TAD ON (TD.CodDisciplina = TAD.fk_CodDisciplina)
GROUP BY    TD.NomeDisciplina
ORDER BY    TD.NomeDisciplina;
```

h)

```
SELECT      TC.NomeCurso, TD.NomeDisciplina, TA.NomeAluno, TAD.NotaMedia
FROM        tabdisciplina AS TD
INNER JOIN  tabalunodisciplina AS TAD ON (TD.CodDisciplina = TAD.fk_CodDisciplina)
INNER JOIN  tabaluno AS TA ON (TA.CodMatricula = TAD.fk_CodMatricula)
INNER JOIN  tabcurso AS TC ON (TA.fk_CodCurso = TC.CodCurso)
ORDER BY    NomeCurso, NomeDisciplina, NomeAluno;
```

i)

```

SELECT TC.NomeCurso, TD.NomeDisciplina, TA.NomeAluno, TAD.NotaMedia
FROM      tabdisciplina      AS TD
INNER JOIN tabalunodisciplina AS TAD ON (TD.CodDisciplina = TAD.fk_CodDisciplina)
INNER JOIN tabaluno          AS TA  ON (TA.CodMatricula  = TAD.fk_CodMatricula)
INNER JOIN tabcurso          AS TC  ON (TA.fk_CodCurso   = TC.CodCurso)
WHERE     (TC.NomeCurso = 'Técnico em Informática' OR
          TC.NomeCurso = 'Técnico em Administração')
ORDER BY NomeDisciplina, NomeAluno;

```

*** Uma Questão sobre Stored Procedure ***

j)

```

DROP PROCEDURE IF EXISTS alunosCurso;
DELIMITER $$
CREATE PROCEDURE alunosCurso (IN nomeCurso VARCHAR(50), OUT notaMedia FLOAT)
BEGIN
    -- Resultado do AVG é atribuído à notaMedia = parâmetro de OUTPUT
    SELECT AVG(TAD.NotaMedia) INTO notaMedia
    FROM      tabdisciplina      AS TD
    INNER JOIN tabalunodisciplina AS TAD ON (TD.CodDisciplina = TAD.fk_CodDisciplina)
    INNER JOIN tabaluno          AS TA  ON (TA.CodMatricula  = TAD.fk_CodMatricula)
    INNER JOIN tabcurso          AS TC  ON (TA.fk_CodCurso   = TC.CodCurso)
    WHERE     TC.NomeCurso      LIKE nomeCurso ;

    -- Ao ser executada, a SP exibe o resultado do SELECT
    SELECT      TC.NomeCurso, TD.NomeDisciplina, TA.NomeAluno, TAD.NotaMedia,
               TAD.NumeroFaltas
    FROM      tabdisciplina      AS TD
    INNER JOIN tabalunodisciplina AS TAD ON (TD.CodDisciplina = TAD.fk_CodDisciplina)
    INNER JOIN tabaluno          AS TA  ON (TA.CodMatricula  = TAD.fk_CodMatricula)
    INNER JOIN tabcurso          AS TC  ON (TA.fk_CodCurso   = TC.CodCurso)
    WHERE     TC.NomeCurso LIKE nomeCurso -- TC.NomeCurso PARECIDO com nomeCurso
    ORDER BY  NomeDisciplina, NomeAluno;
END $$
DELIMITER ;

```

1.

```

SET @notaMediaCurso = 0.0; -- Declara e atribui variável de sessão
SET @nCurso = 'Técnico em Informática'; -- Declara e atribui variável de sessão
CALL alunosCurso (@nCurso, @notaMediaCurso);
SELECT @nCurso AS Curso, @notaMediaCurso AS NotaMediaCurso;

```

2.

```

SET @nCurso = 'Técnico em Eletrônica'; -- Atribui variável de sessão
CALL alunosCurso (@nCurso, @notaMediaCurso);
SELECT @nCurso AS Curso, @notaMediaCurso AS NotaMediaCurso;

```

3.

```

SET @nCurso = 'Técnico em Administração'; -- Atribui variável de sessão
CALL alunosCurso (@nCurso, @notaMediaCurso);
SELECT @nCurso AS Curso, @notaMediaCurso AS NotaMediaCurso;

```

QUESTÃO 2) (2,0 pontos)

- Explique o que é a 3FN para normalização, e como utilizar esse recurso?
- Explique o que é um Trigger SQL e quando ele deve ser utilizado?