TRABALHO DE BANCO DE DADOS –1. BIMESTRE

Grupo: Hicham Karim Jebai, Enzo Manente Ferreira.

Em prova será cobrado os conhecimentos adquiridos neste trabalho.

Peso 4,0 – Entregue em pdf, via TEAMS, os diagramas e SQLs utilizados, bem como as respostas para cada exercício.

Demais itens, serão apresentados no dia.

Este trabalho é composto de 3 partes com diferentes pesos. Entretanto, cada parte não realizada/não entregue implicará em -2,0 pontos na nota final do trabalho –limitado a zerar a nota de entrega.

Apresentar em: 09/10/2023–Todos os alunos devem apresentar e estarem preparados para responder quaisquer dúvidas. Notas podem ser diferenciadas.

PARTE 1 – Peso 1,5

1. Dado a modelagem desnormalizada sobre o estudo de caso Matrícula do Aluno discutida em sala de aula, faça:

   A. Implemente a tabela num SGBD–SOMENTE A TABELA DESNORMALIZADA.
      Para criar a tabela solicitada no SQL foram utilizados os comandos:

```
CREATE TABLE MatriculaAluno (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nomeAluno VARCHAR(255),
    cpfAluno VARCHAR(15),
    foneAluno VARCHAR(15),
    emailAluno VARCHAR(255),
    dtMatriculaAluno DATE,
    nomeCurso VARCHAR(255),
    anoLetivo INT,
    nomeDisciplina VARCHAR(255),
    nomeProfessorDisciplina VARCHAR(255),
    emailProfessor VARCHAR(255)
);
```

```
mysql> show tables;
+----------------------+
| Tables_in_trabalhobd |
+----------------------+
| matriculaaluno       |
+----------------------+
1 row in set (0.00 sec)

mysql> select * from matriculaaluno;
Empty set (0.00 sec)

mysql> desc matriculaaluno;
+--------------------------+--------------+------+-----+---------+----------------+
| Field                    | Type         | Null | Key | Default | Extra          |
+--------------------------+--------------+------+-----+---------+----------------+
| id                       | int          | NO   | PRI | NULL    | auto_increment |
| nomeAluno                | varchar(255) | YES  |     | NULL    |                |
| cpfAluno                 | varchar(15)  | YES  |     | NULL    |                |
| foneAluno                | varchar(15)  | YES  |     | NULL    |                |
| emailAluno               | varchar(255) | YES  |     | NULL    |                |
| dtMatriculaAluno         | date         | YES  |     | NULL    |                |
| nomeCurso                | varchar(255) | YES  |     | NULL    |                |
| anoLetivo                | int          | YES  |     | NULL    |                |
| nomeDisciplina           | varchar(255) | YES  |     | NULL    |                |
| nomeProfessorDisciplina  | varchar(255) | YES  |     | NULL    |                |
| emailProfessor           | varchar(255) | YES  |     | NULL    |                |
+--------------------------+--------------+------+-----+---------+----------------+
11 rows in set (0.00 sec)
```

A imagem representa a tabela no banco de dados após a sua criação. O campo ID foi definido como primary key, além de ser incrementado automaticamente pelo SGBD.

Os inserts no banco de dados foram realizados por meio de comandos via prompt para esta abordagem;

insert into matriculaaluno (nomeAluno, cpfAluno, foneAluno, emailAluno, dtmatriculaAluno, nomeCurso, anoLetivo, nomeDisciplina, nomeProfessorDisciplina, emailProfessor) Values('JoseABC de D', '111.222.333-44', '99999-9999', 'joseabc@xx.com', '2023-10-05', 'ciencia da computacao', 2023, 'Banco de Dados', 'Roberto Gil', 'emailprof@xx.com');

após a realização dos inserts e com a tabela já preenchida com os dados que seriam necessários temos:

```
mysql> select * from matriculaaluno;
+----+--------------+----------------+------------+--------------------+------------------+-----------------------+-----------+----------------+-------------------------+----------------------------+
| id | nomeAluno    | cpfAluno       | foneAluno  | emailAluno         | dtMatriculaAluno | nomeCurso             | anoLetivo | nomeDisciplina | nomeProfessorDisciplina | emailProfessor             |
+----+--------------+----------------+------------+--------------------+------------------+-----------------------+-----------+----------------+-------------------------+----------------------------+
|  1 | JoseABC de D | 111.222.333-44 | 99999-9999 | joseabc@xx.com     | 2023-10-05       | ciencia da computacao |      2023 | Banco de Dados | Roberto Gil             | emailprof@xx.com           |
|  2 | Hicham       | 222.333.444-55 | 99999-8888 | hicham@xx.com      | 2023-10-05       | ciencia da computacao |      2023 | BD             | Roberto Gil             | emailprof@xx.com           |
|  3 | Hicham       | 222.333.444-55 | 99999-8888 | hicham@xx.com      | 2023-10-05       | ciencia da computacao |      2023 | Redes          | profRedes               | emailprofredes@xx.com      |
|  4 | Hicham       | 222.333.444-55 | 99999-8888 | hicham@xx.com      | 2023-10-05       | ciencia da computacao |      2023 | Eng Soft 1     | profEngSoft1            | emailprofEngSoft@xx.com    |
|  5 | JoseABC de D | 111.222.333-44 | 99999-9999 | joseabc@xx.com     | 2023-10-05       | ciencia da computacao |      2023 | Comp Grafica   | profCompGrafica         | emailprofCompGrafica@xx.com |
|  6 | JoseABC de D | 111.222.333-44 | 99999-9999 | joseabc@xx.com     | 2022-10-05       | ciencia da computacao |      2023 | Redes          | profRedes               | emailprofredes@xx.com      |
|  7 | Enzo         | 333.444.555-66 | 98888-7777 | Enzo@xx.com        | 2023-10-01       | ciencia da computacao |      2023 | Banco de Dados | Roberto Gil             | emailprof@xx.com           |
|  8 | Enzo         | 333.444.555-66 | 98888-7777 | Enzo@xx.com        | 2021-10-01       | ciencia da computacao |      2023 | Redes          | profRedes               | emailprofredes@xx.com      |
|  9 | Enzo         | 333.444.555-66 | 98888-7777 | Enzo@xx.com        | 2021-10-01       | ciencia da computacao |      2021 | computaçao 1   | profComp1               | emailprofComp1@xx.com      |
| 10 | Guilherme L. | 444.333.555-66 | 96666-6666 | GuilhermeL@xx.com  | 2020-01-10       | Engenharia Mecanica   |      2021 | Calculo 1      | profCalculo1            | emailprofcalculo1@xx.com   |
| 11 | Guilherme L. | 444.333.555-66 | 96666-6666 | GuilhermeL@xx.com  | 2020-01-10       | Engenharia Mecanica   |      2021 | Calculo 2      | profCalculo2            | emailprofcalculo2@xx.com   |
| 12 | Guilherme L. | 444.333.555-66 | 96666-6666 | GuilhermeL@xx.com  | 2020-01-10       | Engenharia Mecanica   |      2021 | Fisica2        | profFisica2             | emailproffisica2@xx.com    |
+----+--------------+----------------+------------+--------------------+------------------+-----------------------+-----------+----------------+-------------------------+----------------------------+
12 rows in set (0.00 sec)
```

B. Crie um programa em JAVA para conexão com o SGBD e realize os cenários descritos nos próximos itens –não altere a sequência de execução;

Temos a conexão do SGBD com o programa em java, para realizar os próximos cenários a solução escolhida foi mostrar os resultados no prompt por meio de System.out.println(“”);

```java
public class ConexaoMySQL {
    public static void main(String[] args) {
        // Configurações de conexão com o banco de dados
        String url = "jdbc:mysql://localhost:3306/trabalhobd"; // Altere para a URL do seu banco de dados
        String usuario = "User"; // Altere para o usuário do seu banco de dados
        String senha = "Senha"; // Altere para a senha do seu banco de dados

        try {
            // Conectar ao banco de dados
            Connection conexao = DriverManager.getConnection(url, usuario, senha);

            // Criar uma declaração SQL
            Statement statement = conexao.createStatement();
```

C. Considere uma maneira do sistema mostrar ao futuro alunos quais os cursos disponíveis;
Para este item obtivemos a seguinte solução:

```java
// Consulta para obter os cursos disponíveis
String consultaCursos = "SELECT DISTINCT nomeCurso FROM MatriculaAluno";
ResultSet resultSetCursos = statement.executeQuery(consultaCursos);

// Imprimir os cursos disponíveis
System.out.println("Cursos Disponíveis:");
while (resultSetCursos.next()) {
    String curso = resultSetCursos.getString( columnLabel: "nomeCurso");
    System.out.println(curso);
}
```

D. Considere uma maneira do sistema mostrar ao futuro aluno quais as disciplinas disponíveis para um determinado curso;
Para este item obtivemos a seguinte solução:

```java
// Consulta para obter as disciplinas disponíveis para cada curso
String consultaDisciplinas = "SELECT DISTINCT nomeCurso, nomeDisciplina FROM MatriculaAluno";
ResultSet resultSetDisciplinas = statement.executeQuery(consultaDisciplinas);

// Imprimir as disciplinas disponíveis para cada curso
System.out.println("\nDisciplinas Disponíveis por Curso:");
while (resultSetDisciplinas.next()) {
    String curso = resultSetDisciplinas.getString( columnLabel: "nomeCurso");
    String disciplina = resultSetDisciplinas.getString( columnLabel: "nomeDisciplina");
    System.out.println("Curso: " + curso + ", Disciplina: " + disciplina);
}
```

E. Considere uma maneira do sistema mostrar quais os alunos já cadastrados, porém ainda não matriculados em nenhum curso;
Para este item obtivemos a seguinte solução:

```java
// Consulta para obter os alunos que não estão matriculados em nenhum curso
String consultaAlunosNaoMatriculados = "SELECT DISTINCT nomeAluno FROM MatriculaAluno WHERE nomeCurso IS NULL";
ResultSet resultSetAlunosNaoMatriculados = statement.executeQuery(consultaAlunosNaoMatriculados);

// Imprimir os alunos não matriculados em nenhum curso
System.out.println("Alunos não matriculados em nenhum curso:");
while (resultSetAlunosNaoMatriculados.next()) {
    String aluno = resultSetAlunosNaoMatriculados.getString( columnLabel: "nomeAluno");
    System.out.println(aluno);
}
```

F. Matrícula 4 alunos diferentes, e para cada aluno, considere de três a quatro disciplinas por aluno;
Após a inserção dos alunos na tabela temos:

```
mysql> select * from matriculaaluno;
+----+-------------+-------------+-------------+-----------------+-----------------+-----------------------+-----------+-----------------+-------------------------+-----------------------+
| id | nomeAluno   | cpfAluno    | foneAluno   | emailAluno      | dtMatriculaAluno | nomeCurso             | anoLetivo | nomeDisciplina  | nomeProfessorDisciplina | emailProfessor        |
+----+-------------+-------------+-------------+-----------------+-----------------+-----------------------+-----------+-----------------+-------------------------+-----------------------+
|  1 | JoseABC de D | 111.222.333-44 | 99999-9999 | joseabc@xx.com  | 2023-10-05      | ciencia da computacao |      2023 | Banco de Dados  | Roberto Gil             | emailprof@xx.com      |
|  2 | Hicham      | 222.333.444-55 | 99999-8888 | hicham@xx.com   | 2023-10-05      | ciencia da computacao |      2023 | BD              | Roberto Gil             | emailprof@xx.com      |
|  3 | Hicham      | 222.333.444-55 | 99999-8888 | hicham@xx.com   | 2023-10-05      | ciencia da computacao |      2023 | Redes           | profRedes               | emailprofredes@xx.com |
|  4 | Hicham      | 222.333.444-55 | 99999-8888 | hicham@xx.com   | 2023-10-05      | ciencia da computacao |      2023 | Eng Soft 1      | profEngSoft1            | emailprofEngSoft@xx.com |
|  5 | JoseABC de D | 111.222.333-44 | 99999-9999 | joseabc@xx.com  | 2023-10-05      | ciencia da computacao |      2023 | Comp Grafica    | profCompGrafica         | emailprofCompGrafica@xx.com |
|  6 | JoseABC de D | 111.222.333-44 | 99999-9999 | joseabc@xx.com  | 2022-10-05      | ciencia da computacao |      2023 | Redes           | profRedes               | emailprofredes@xx.com |
|  7 | Enzo        | 333.444.555-66 | 98888-7777 | Enzo@xx.com     | 2023-10-01      | ciencia da computacao |      2023 | Banco de Dados  | Roberto Gil             | emailprof@xx.com      |
|  8 | Enzo        | 333.444.555-66 | 98888-7777 | Enzo@xx.com     | 2021-10-01      | ciencia da computacao |      2023 | Redes           | profRedes               | emailprofredes@xx.com |
|  9 | Enzo        | 333.444.555-66 | 98888-7777 | Enzo@xx.com     | 2021-10-01      | ciencia da computacao |      2021 | computaçao 1    | profComp1               | emailprofComp1@xx.com |
| 10 | Guilherme L. | 444.333.555-66 | 96666-6666 | GuilhermeL@xx.com | 2020-01-10      | Engenharia Mecanica   |      2021 | Calculo 1       | profCalculo1            | emailprofcalculo1@xx.com |
| 11 | Guilherme L. | 444.333.555-66 | 96666-6666 | GuilhermeL@xx.com | 2020-01-10      | Engenharia Mecanica   |      2021 | Calculo 2       | profCalculo2            | emailprofcalculo2@xx.com |
| 12 | Guilherme L. | 444.333.555-66 | 96666-6666 | GuilhermeL@xx.com | 2020-01-10      | Engenharia Mecanica   |      2021 | Fisica2         | profFisica2             | emailproffisica2@xx.com |
+----+-------------+-------------+-------------+-----------------+-----------------+-----------------------+-----------+-----------------+-------------------------+-----------------------+
12 rows in set (0.00 sec)
```

G. Nos quatro alunos, considere que pelo menos dois deles estarão sendo matriculados na disciplina de Banco de Dados. Porém no nome da disciplina, para um insira como “Banco de Dados” e para o outro “BD”;

Este item foi realizado por meio de inserts no prompt do SGBD, os comandos utilizados foram:

```
insert into matriculaaluno (nomeAluno, cpfAluno, foneAluno, emailAluno, dtmatriculaAluno, nomeCurso, anoLetivo, nomeDisciplina, nomeProfessorDisciplina, emailProfessor) Values('JoseABC de D', '111.222.333-44', '99999-9999', 'joseabc@xx.com', '2023-10-05', 'ciencia da computacao', 2023, 'Banco de Dados', 'Roberto Gil', 'emailprof@xx.com');
```

```
insert into matriculaaluno (nomeAluno, cpfAluno, foneAluno, emailAluno, dtmatriculaAluno, nomeCurso, anoLetivo, nomeDisciplina, nomeProfessorDisciplina, emailProfessor) Values('Hicham', '222.333.444-55', '99999-8888', 'hicham@xx.com', '2023-10-05', 'ciencia da computacao', 2023, 'BD', 'Roberto Gil', 'emailprof@xx.com');
```

H. Um dos alunos deve se chamar “José ABC de D”;
Este item foi realizado através de um insert no prompt do SGBD:

```
insert into matriculaaluno (nomeAluno, cpfAluno, foneAluno, emailAluno, dtmatriculaAluno, nomeCurso, anoLetivo, nomeDisciplina, nomeProfessorDisciplina, emailProfessor) Values('JoseABC de D', '111.222.333-44', '99999-9999', 'joseabc@xx.com', '2023-10-05', 'ciencia da computacao', 2023, 'Banco de Dados', 'Roberto Gil', 'emailprof@xx.com');
```

Aqui estão mais algumas disciplinas cursadas por “JoseABC de D”:

5	JoseABC de D	111.222.333-44	99999-9999	joseabc@xx.com	2023-10-05	ciencia da computacao	2023	Comp Grafica	profCompGrafica	emailprofCompGrafica@xx.com
6	JoseABC de D	111.222.333-44	99999-9999	joseabc@xx.com	2022-10-05	ciencia da computacao	2023	Redes	profRedes	emailprofredes@xx.com

I. . Implemente uma interface qualquer em java (pode ser apenas utilizando System.out.println(“”) para mostrar uma matrícula do aluno realizada; Interface utilizando System.out.println(“”); para mostrar os dados de uma matrícula realizada do aluno:

```
// Consulta para obter informações do aluno pelo nome
String consultaAlunoPorNome = "SELECT nomeAluno, nomeCurso, nomeDisciplina, cpfAluno, foneAluno FROM MatriculaAluno WHERE nomeAluno = ?";
PreparedStatement preparedStatement = conexao.prepareStatement(consultaAlunoPorNome);
preparedStatement.setString( parameterIndex: 1, nomeAluno);

ResultSet resultSetAluno = ((PreparedStatement) preparedStatement).executeQuery();

// Imprimir as informações do aluno
System.out.println("\nInformações do Aluno:");
while (resultSetAluno.next()) {
    String aluno = resultSetAluno.getString( columnLabel: "nomeAluno");
    String curso = resultSetAluno.getString( columnLabel: "nomeCurso");
    String disciplina = resultSetAluno.getString( columnLabel: "nomeDisciplina");
    String cpf = resultSetAluno.getString( columnLabel: "cpfAluno");
    String telefone = resultSetAluno.getString( columnLabel: "foneAluno");
    System.out.println("Aluno: " + aluno + ", Curso: " + curso + ", Disciplina: " + disciplina + ", Cpf: " + cpf +", telefone: " +telefone);
}
```

Resultados da pesquisa para o aluno: Enzo:

```
Digite o nome do aluno que deseja buscar: Enzo

Informações do Aluno:
Aluno: Enzo, Curso: ciencia da computacao, Disciplina: Banco de Dados, Cpf: 333.444.555-66, telefone: 98888-7777
Aluno: Enzo, Curso: ciencia da computacao, Disciplina: Redes, Cpf: 333.444.555-66, telefone: 98888-7777
Aluno: Enzo, Curso: ciencia da computacao, Disciplina: computaçao 1, Cpf: 333.444.555-66, telefone: 98888-7777
```

J. Liste todos os alunos que estão matriculados na disciplina de Banco de Dados. Alguém não apareceu? Por quê?
Ao listar todos os alunos matriculados em Bando de dados, temos:

```
Alunos matriculados na disciplina de Banco de Dados:
JoseABC de D
Enzo
```

O aluno Hicham não aparece na pesquisa da disciplina de banco de dados, pois no SGBD o nome da disciplina na matrícula foi feito como ‘BD’, quando realizamos a busca apenas as matriculas com “Banco de Dados” são exibidas.

PARTE 2 –Peso 0,5

A partir do estudo de caso normalizado do trabalho 1, considere em seu cadastro, um aluno matriculado em 2015 na disciplina de Banco de Dados, onde o professor foi o “João ABCD” e outro aluno matriculado em 2023 onde o professor foi o “Roberto Gil”.

Baseado nos dados acima, implemente o relacionamento entre disciplina e professor comas seguintes variantes:

a) Proposta de Solução 1:

Destacamos aqui que o modelo proposto permite que uma disciplina tenha apenas um professor, uma vez que uma chave estrangeira para o professor é mantida na tabela da disciplina.

Implementação em SQL dessa proposta:

```
CREATE TABLE Professor (
   codProfessor INT PRIMARY KEY,
   nomeProfessor VARCHAR(255)
);
CREATE TABLE Disciplina (
   codDisciplina INT PRIMARY KEY,
   nomeDisciplina VARCHAR(255),
   codProfessor INT,
   FOREIGN KEY (codProfessor) REFERENCES Professor(codProfessor)
);
```

Impacto:

Esse modelo é simples, mas limita a flexibilidade para expressar as relações entre as entidades. Ele não suporta o conceito de uma disciplina sendo lecionada por vários professores diferentes em anos letivos diferentes.

a) Proposta de Solução 2

Este modelo propõe a utilização de uma tabela de junção (DisciplinaProfessor) para gerenciar as múltiplas relações entre professores e disciplinas. Isso permite uma flexibilidade muito maior, em que uma disciplina pode ser lecionada por diferentes professores em diferentes anos letivos.

Implementação em SQL dessa proposta:

```
CREATE TABLE Professor (
   codProfessor INT PRIMARY KEY,
   nomeProfessor VARCHAR(255)
);
CREATE TABLE Disciplina (
   codDisciplina INT PRIMARY KEY,
   nomeDisciplina VARCHAR(255)
);
CREATE TABLE DisciplinaProfessor (
   codDisciplina INT,
   codProfessor INT,
   anoLetivo INT,
   dtinicioNaDisciplina DATE,
   PRIMARY KEY(codDisciplina, codProfessor, anoLetivo, dtinicioNaDisciplina),
   FOREIGN KEY (codDisciplina) REFERENCES Disciplina(codDisciplina),
   FOREIGN KEY (codProfessor) REFERENCES Professor(codProfessor)
);
```

Impacto:

Este modelo permite uma grande flexibilidade e acomoda muitos casos do mundo real, pois uma disciplina pode ser lecionada por diferentes professores em diferentes anos letivos. No entanto, o gerenciamento dos dados é um pouco mais complexo e requer mais consultas inter-tabelas, o que pode impactar no desempenho se as tabelas se tornarem muito grandes.

PARTE 3 –Peso 2,0

Considerando a modelagem na 3.FN já apresentado em sala de aula para matricula de Aluno, realize os seguintes ajustes:

a) Considerar Endereço completo do Aluno e professor
   Para o endereço de aluno e professor temos: as tabelas

```
CREATE TABLE Endereço (
   idEndereço INT AUTO_INCREMENT PRIMARY KEY,
   Cep VARCHAR(8),
   idProfessor INT,
   idBairro INT,
   idcidade INT,
   FOREIGN KEY (idProfessor) REFERENCES Professor(idProfessor),
   FOREIGN KEY (idBairro) REFERENCES Bairro (idBairro),
   FOREIGN KEY (idCidade) REFERENCES Cidade(idCidade)
);


CREATE TABLE Bairro(
  idBairro INT AUTO_INCREMENT PRIMARY KEY,
  nomeBairro VARCHAR(45) NULL
);

CREATE TABLE Cidade (
  idCidade INT AUTO_INCREMENT PRIMARY KEY,
  nomeCidade VARCHAR(45) NULL
);
```

b) Tanto para aluno, quanto para professor, registrar seus fones e e-mails
   Para registrar fone de Aluno e professor temos:

```
CREATE TABLE Professor (
  idProfessor INT AUTO_INCREMENT PRIMARY KEY,
  nomeProfessor VARCHAR(45),
  emailProfessor VARCHAR(45),
  foneProfessor VARCHAR(45));
CREATE TABLE Aluno(
  idAluno INT AUTO_INCREMENT PRIMARY KEY,
  nomeAluno VARCHAR(45),
  cpfAluno VARCHAR(45),
  foneAluno VARCHAR(45),
  emailAluno VARCHAR(45),
  idEndereço INT,
  FOREIGN KEY (idEndereço) REFERENCES Endereço(idEndereço));
```

c)No momento da matrícula no Ano Letivo o sistema deve validar:

•Se o aluno já está matriculado no Curso informado;

•Se o Ano Letivo é válido. Por exemplo, querer matricular no ano letivo de 2024.

Para a validação do ano letivo, foi implementado no SGBD:

```
DELIMITER //

CREATE TRIGGER valida_matricula

BEFORE INSERT ON MatriculaAluno

FOR EACH ROW

BEGIN

    DECLARE num_matriculas INT;


    -- Verificar se o aluno já está matriculado no mesmo curso e ano letivo

    SELECT COUNT(*) INTO num_matriculas

    FROM MatriculaAluno

    WHERE idAluno = NEW.idAluno

    AND idCurso = NEW.idCurso

    AND YEAR(anoLetivo) = YEAR(NEW.anoLetivo);


    IF num_matriculas > 0 THEN

        SIGNAL SQLSTATE '45000'

        SET MESSAGE_TEXT = 'Aluno já está matriculado no mesmo curso no ano letivo informado';

    END IF;

END;

//

DELIMITER ;
```

d) Apresente via PDF o modelo E-R:
   Se encontra no Arquivo MODELO_E-R;

e) Implemente todas as tabelas e escreva um programa que se conecte ao SGBD, escrevendo lógicas e consultas que atendam as questões abaixo:
Para este item foram criada as seguintes tabelas:

```
+----------------------+
| Tables_in_trabbdpt3  |
+----------------------+
| aluno                |
| bairro               |
| cidade               |
| curso                |
| disciplina           |
| endereço             |
| matriculaaluno       |
| matriculadisciplina  |
| professor            |
+----------------------+
```

f) Considere uma maneira do sistema mostrar ao futuro alunos quais os cursos disponíveis
Função e comando para pegar todos os cursos disponíveis e imprimir:

```java
//Comando para pegar todos os cursos
String sqlGetCoursos = "SELECT nomeCurso FROM Curso";
Statement statement1 = conexao.createStatement();
ResultSet coursos = statement1.executeQuery(sqlGetCoursos);
System.out.println("Cursos disponíveis:");

//Imprimir todos os cursos no console
while (coursos.next()) {
    System.out.println(coursos.getString( columnLabel: "nomeCurso"));
}
```

g) Considere uma maneira do sistema mostrar ao futuro aluno quais as disciplinas disponíveis para um determinado curso
Função e comandos para pegar e imprimir todos as disciplinas de um curso;

```java
//Comando para pegar todas as disciplinas de um curso - suponho que seja o curso com id=1
String sqlGetDisciplinas = "SELECT Disciplina.nomeDisciplina FROM Disciplina JOIN MatriculaDisciplina ON Disciplina.idDisciplina = MatriculaDisciplina.idDisciplina JOIN MatriculaAluno ON
Statement statement2 = conexao.createStatement();
ResultSet disciplinas = statement2.executeQuery(sqlGetDisciplinas);
System.out.println("\nDisciplinas disponiveis no curso selecionado:");

//Imprimir todas as disciplinas no console
while (disciplinas.next()) {
    System.out.println(disciplinas.getString( columnLabel: "nomeDisciplina"));
}
```

h) Considere uma maneira do sistema mostrar quais os alunos já cadastrados, porém ainda não matriculados em nenhum curso

```java
// Comando para pegar todos os alunos não matriculados em nenhum curso
String sqlGetalunosnaomatriculados = "SELECT nomeAluno FROM Aluno LEFT JOIN MatriculaAluno ON Aluno.idAluno = MatriculaAluno.idAluno WHERE MatriculaAluno.idAluno IS NULL";
Statement statement3 = conexao.createStatement();
ResultSet alunosnaomatriculados = statement3.executeQuery(sqlGetalunosnaomatriculados);
System.out.println("\nAlunos ainda não matriculados em nenhum curso:");

while (alunosnaomatriculados.next()) {
    System.out.println(alunosnaomatriculados.getString( columnLabel: "nomeAluno"));
}
```

i) Matrícula 4 alunos em dois cursos diferentes

j) Liste quais alunos estão num curso e em outro

k) Para cada aluno matriculado num curso, matricule os no ano letivo de 2023, sendo que cada aluno esteja em algumas disciplinas

l) Liste os alunos matriculados no curso de Ciência da Computação que fazem a disciplina de Banco de Dados em relação ao realizado na PARTE1:

m) O que se observou de diferente desta vez, do ponto de vista da implementação (Java / SQL)?

n) O que você observou de diferente entre os dois modelos de dados, considerando as possíveis funcionalidades e requisitos do futuro software?