		<b>Prova de Bancos de Dados</b>	
		<b>Professor:</b> Gleison Brito	
<b>Turma:</b> BHZ231BMTDS		<b>Data:</b>	
<b>Aluno(a):</b>			<b>Número:</b>
<i>As respostas deverão ser redigidas exclusivamente com caneta azul ou preta. Respostas a lápis não serão revisadas em hipótese alguma. Respostas com rasuras não serão avaliadas. Não use corretivo. Telefones celulares, agendas, estojos, bolsa e afins não podem ficar sobre as carteiras.</i>			<b>VALOR:</b> 3,0 PONTOS  <b>NOTA:</b>

### Prova de Bancos de Dados

1) Um Sistema de Gerenciamento de Banco de Dados (SGBD) é responsável pelo armazenamento, recuperação e manipulação de dados em um banco de dados. Marque "V" para verdadeiro e "F" para falso para as seguintes afirmações relacionadas ao SGBD: (0.3 pt)

(V) Um SGBD permite o compartilhamento simultâneo de dados entre vários usuários e aplicativos.

(F) Um SGBD não oferece segurança para proteger os dados armazenados no banco de dados.

(V) Um SGBD fornece mecanismos de consulta que permitem aos usuários recuperar informações de maneira eficiente.

(F) Os SGBDs não oferecem suporte para transações, tornando impossível a manutenção de consistência nos dados.

2) O dicionário de dados desempenha um papel fundamental no gerenciamento de um Sistema de Gerenciamento de Banco de Dados (SGBD). Marque "V" para verdadeiro e "F" para falso para as seguintes afirmações relacionadas a dicionário de dados: (0.3 pt)

(V) O dicionário de dados armazena informações sobre a estrutura do banco de dados, como tabelas, campos, tipos de dados e relacionamentos.

(F) O dicionário de dados não é acessível aos administradores do banco de dados e aos desenvolvedores de aplicativos.

(V) As informações no dicionário de dados são usadas para garantir a integridade dos dados e a consistência das operações no banco de dados.

(F) A atualização do dicionário de dados é realizada manualmente e não está relacionada a operações automáticas no banco de dados.

3) O SQL (Structured Query Language) é uma linguagem de programação usada para gerenciar e consultar bancos de dados relacionais. Marque "V" para verdadeiro e "F" para falso para as seguintes afirmações relacionadas ao SQL: (0.3 pt)

(F) O SQL é uma linguagem de programação de propósito geral usada para criar aplicativos de desktop e web.

(V) O SQL é amplamente utilizado para criar, consultar e modificar bancos de dados relacionais.

(V) A linguagem SQL inclui instruções para recuperar dados de um banco de dados, como SELECT, e também para atualizar, inserir e excluir registros.

(F) O SQL não é padronizado, o que significa que a sintaxe e as funcionalidades podem variar significativamente entre diferentes sistemas de gerenciamento de banco de dados.

4) Qual das seguintes afirmações descreve corretamente o conceito de chave primária (primary key) em um banco de dados? (0.3 pt)

( ) A chave primária é uma coluna que pode conter valores duplicados e é usada para identificar registros exclusivos em uma tabela.

( ) A chave primária é uma coluna que contém valores únicos e é usada para identificar registros duplicados em uma tabela.

(X) A chave primária é uma coluna ou um conjunto de colunas que contém valores únicos e é usada para identificar registros exclusivos em uma tabela.

( ) A chave primária é uma coluna que não é necessária em uma tabela, pois todas as tabelas de banco de dados podem funcionar sem uma chave primária.

5) Em um banco de dados relacional, qual das seguintes afirmações é verdadeira em relação ao relacionamento e à chave estrangeira (foreign key) (0.3 pt)

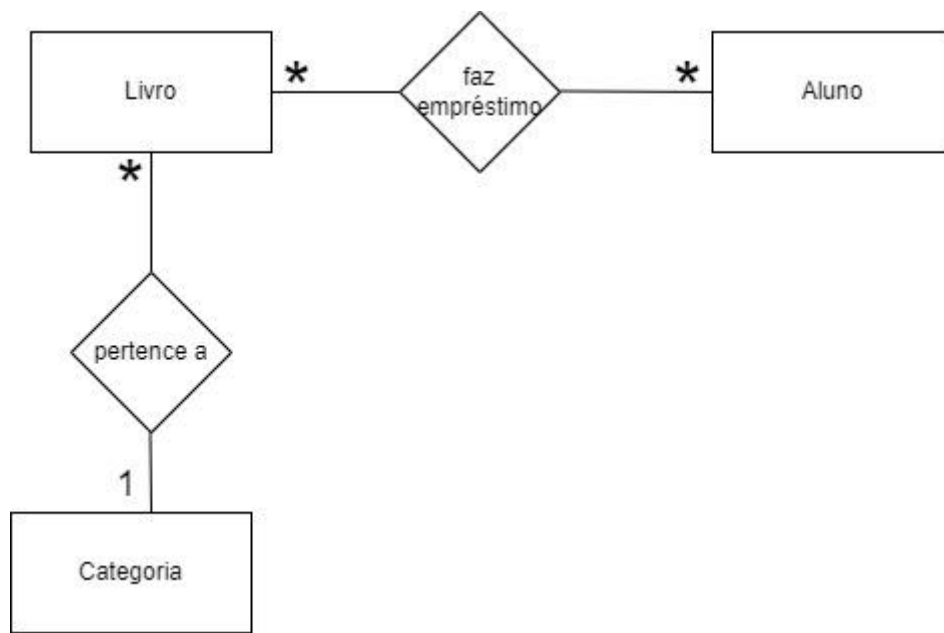
( ) A chave estrangeira é usada apenas para identificar registros únicos em uma tabela.

( ) A chave estrangeira é usada para garantir que os registros em uma tabela não tenham duplicatas e podem ser usados para criar uma tabela independente sem relacionamentos

(X) A chave estrangeira é uma coluna em uma tabela que faz referência a uma chave primária em outra tabela, estabelecendo um vínculo entre as duas tabelas.

( ) A chave estrangeira é usada para criar uma cópia de uma tabela em outra tabela, mantendo os dados completamente independentes.

6) O diagrama entidade-relacionamento (DER) a seguir representa uma biblioteca escolar com as entidades "Livro", "Categoria" e "Aluno". Defina a cardinalidade dos relacionamentos. No modelo um livro pertence a uma categoria, mas uma categoria pode ter vários livros; cada aluno pode fazer empréstimo de vários livros e cada livro pode ser emprestado por vários alunos. (0.3 pt)



7) Desenhe um diagrama entidade-relacionamento (DER) para o banco de dados de um aplicativo de contatos, considerando as seguintes entidades e relacionamentos. (0.4 pt)

**Entidades:**

**Contato** - Atributos: ID (chave primária), Nome, Sobrenome, Data de Nascimento.

**Telefone** - Atributos: ID (chave primária), Número de Telefone, Tipo (por exemplo, Celular, Residencial).

**Grupo** - Atributos: ID (chave primária), Nome do Grupo, Descrição.

**Email** - Atributos: ID (chave primária), Endereço de Email, Tipo (por exemplo, Pessoal, Trabalho).

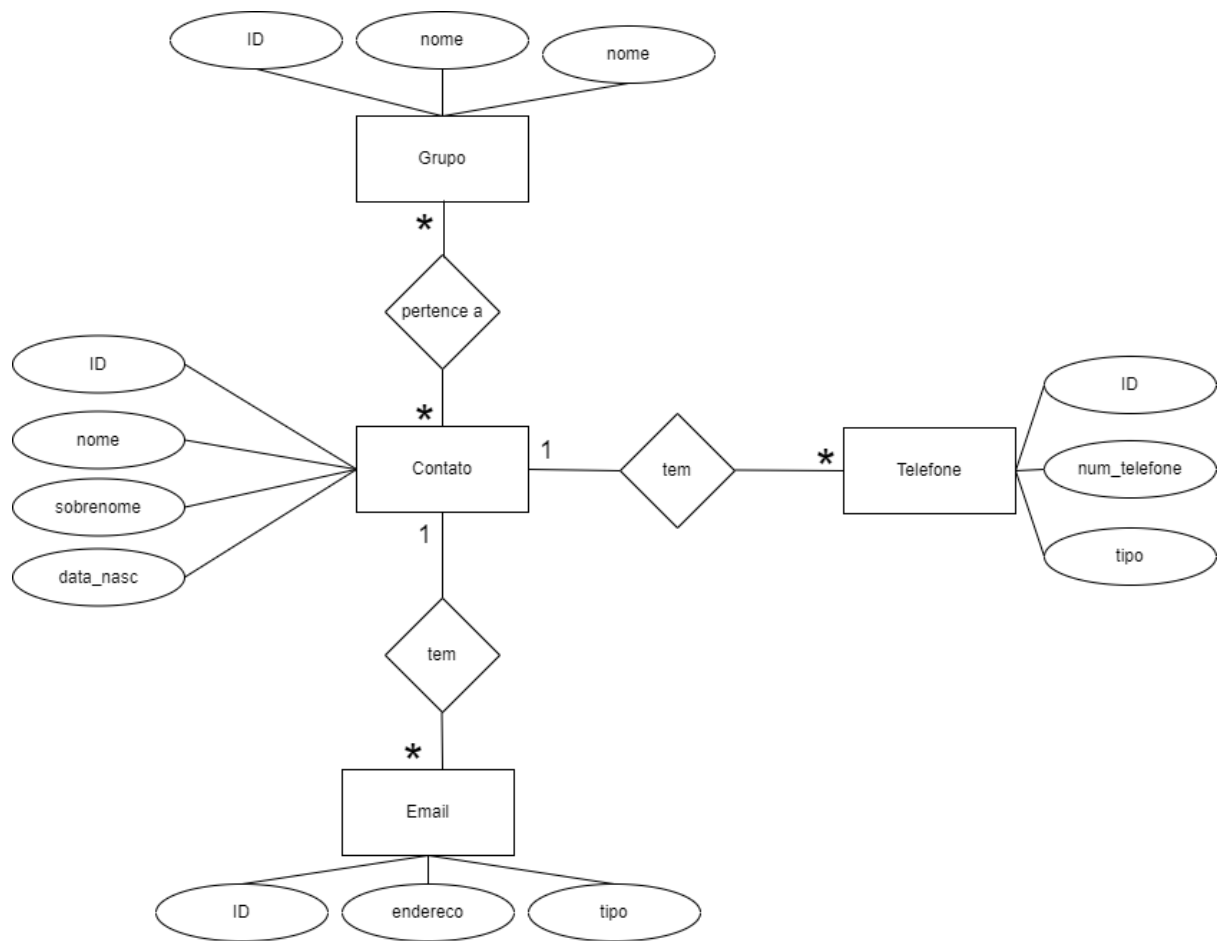
**Relacionamentos:**

Cada "Contato" pode ter um ou mais números de telefone associados a ele.

Cada "Contato" pode ser membro de zero ou mais "Grupos" (por exemplo, Família, Amigos, Trabalho).

Cada "Grupo" pode conter vários "Contatos".

Cada "Contato" pode ter um ou mais endereços de email associados a ele.



8) Escreva uma consulta na linguagem SQL para criar as entidades Jogador e Time no banco de dados dbBrasileirao (0.4 pt).

A tabela Jogador deve conter os atributos ID (chave primária), nome, dataDeNascimento, IDTime (chave estrangeira); A tabela Time deve conter os atributos ID (chave primária), nome, cidade, tecnico.

Modelo para criação de tabela:

```

CREATE TABLE <nome da tabela> ( ID INT PRIMARY KEY,
<nome_do_atributo> tipo,
<nome_do_atributo> tipo,
<nome_do_atributo_chave_estrangeira> tipo,
FOREIGN KEY (<nome_do_atributo_chave_estrangeira>) REFERENCES
<Nome_Da_Tabela_Da_Chave_Estrangeira>(<nome_do_atributo_utilizado_como_chave_estrangeira>);
  
```

```
CREATE TABLE Time ( ID INT PRIMARY KEY,
nome varchar(255),
cidade varchar(255),
tecnico varchar(255)
);
```

```
CREATE TABLE Jogador ( ID INT PRIMARY KEY,
nome varchar(255),
dataDeNascimento varchar(255),
IDTime tipo,
FOREIGN KEY (IDTime) REFERENCES Time (ID) );
```

- 9) Suponha uma tabela chamada Pessoa com os atributos ID, nome e idade. Dada a consulta

```
SELECT nome, idade FROM pessoas WHERE idade > 18;
```

Descreva a consulta, explicando os comandos **SELECT**, **FROM** e **WHERE**. (0.2 pt)

Seleciona os campos nome e idade (select) da tabela pessoas (from), filtrando apenas os registros onde a idade é maior que 18 (where).

- 10) Suponha que temos duas tabelas em nosso banco de dados, 'Clientes' e 'Pedidos', e queremos recuperar os nomes dos clientes que fizeram pedidos. (0.2)

A tabela Clientes contém os registros:

ClienteID	Nome
1	Cliente A
2	Cliente B
3	Cliente C

A tabela Pedidos contém os registros:

PedidoID	ClienteID	Data
101	1	2023-01-15
102	2	2023-01-16
103	1	2023-01-17
104	3	2023-01-18

Desenhe a tabela com o resultado da consulta:

```
SELECT Clientes.Nome, Pedidos.Data FROM Clientes INNER JOIN
Pedidos ON Clientes.ClienteID = Pedidos.ClienteID;
```

Nome	Data
Cliente A	2023-01-15
Cliente B	2023-01-16
Cliente A	2023-01-17
Cliente B	2023-01-18

**ESPAÇO PARA RESOLUÇÃO DAS QUESTÕES**

**ESPAÇO PARA RESOLUÇÃO DAS QUESTÕES**