





Universidade Federal Rural de Pernambuco  
Departamento de Estatística e Informática - DEINFO  
Bacharelado em Sistemas de Informação  
Disciplina: Introdução ao Armazenamento e Análise de Dados - IAAD  
Professora: Roberta Macêdo Marques Gouveia

**Atividade prática utilizando o Banco de Dados Relacional MySQL**  
(Junções, Consultas Aninhadas, Funções de Agregação, Agrupamento,  
*Views, Triggers, Stored Procedures, Álgebra Relacional e Engenharia Reversa*)

**Estudante (nome completo): Estom Paulino da Silva Junior**

 Pontuação da atividade: 10 pontos

 Prazo: (23/08/2022)

⚠ **Atenção:** está ativada a verificação de plágio (originalidade) do Google Classroom. As questões com respostas iguais entre estudantes ou copiadas da internet serão desconsideradas/anuladas.

**QUESTÃO 1 (8,0 pontos)**

Considere o esquema do banco de dados relacional “Clínicas Médicas” apresentado abaixo:

```
/* tabela com informações de clínicas médicas */  
Clinica (CodCli, NomeCli, Endereco, Telefone, E-mail)  
  
/* tabela com informações de médicos */  
Medico (CodMed, NomeMed, Genero, Telefone, E-mail, CodEspec)  
CodEspec referencia Especialidade  
  
/* tabela com informações de pacientes */  
Paciente (CpfPaciente, NomePac, DataNascimento, Genero, Telefone, E-mail)  
  
/* tabela que relaciona médicos e clínicas */  
ClinicaMedico (CodCli, CodMed, DataIngresso, CargaHorariaSemanal)  
CodCli referencia Clinica  
CodMed referencia Medico  
  
/* tabela com informações da agenda de consultas dos médicos */  
AgendaConsulta (CodCli, CodMed, CpfPaciente, Data/Hora)  
(CodCli, CodMed) referencia ClinicaMedico  
CpfPaciente referencia Paciente  
  
/* tabela com informações de especialidades médicas */  
Especialidade (CodEspec, NomeEspec, Descricao)
```

**A)** Escreva as instruções DDL apropriadas para criação das 6 tabelas do banco de dados Clínicas Médicas (**apresente o link de seu Google Drive contendo o script SQL**). Lembre-se de especificar as chaves primárias e estrangeiras. Assuma as seguintes restrições de integridade:

- Os e-mails dos(as) médicos(as) e pacientes têm que ser únicos, ou seja, não podem ter dois ou mais médicos(as) com o mesmo e-mail, assim como não podem ter dois ou mais pacientes com o mesmo e-mail.
- O atributo *DataIngresso* é do tipo/domínio DATE. Os tipos dos demais atributos podem ser escolhidos livremente.
- Os nomes dos(as) médicos(as) e pacientes não podem ser nulos, além da descrição da especialidade.

- Ao excluir uma especialidade, todos os médicos(as) associados(as) a essa especialidade devem ser excluídos(as) automaticamente. Além disso, ao alterar o código de uma especialidade, todos(as) os(as) médicos(as) que estão associados(as) a este código devem ter seus códigos de especialidade alterados automaticamente.
- Um(a) paciente não pode ser excluído(a) caso exista alguma consulta associada a ele(a).

Dica: utilizar as ações de disparo referencial '*on delete*' e/ou '*on update*'.

<https://github.com/EstomJr/IAAD>

**B)** Popule as 6 tabelas do banco de dados Clínicas Médicas. Apresente o *link de seu Google Drive* contendo o *script* com os comandos *insert* (no script deve-se constar apenas os comandos *insert*).

<https://github.com/EstomJr/IAAD>

**C)** Apresente exemplos de comandos SQL (podendo ser inserção, remoção ou atualização) que constatem que as restrições de integridade implementadas na letra “A” estão sendo aplicadas corretamente.

```
insert into Medicos values
(20001, null, 'M', '81997802526', 'Estom@mail.com', 80001)
```

Error Code Column 'NomeMed' cannot be null

**D)** Especifique o comando SQL que retorne os nomes dos(as) médicos(as) e os nomes das respectivas especialidades em que eles(as) atuam. Apresentar a consulta de três formas: (I) junção na cláusula *where*, (II) conexão interna (*inner join*), (III) junção natural (*natural join*).

```
select p.NomeMed, s.NomeEspec
from Medicos as p, Especialidade as s
where p.CodEspec = s.CodEspec;
```

```
select Medico.NomeMed, Especialidade.NomeEspec
from Medico inner join Especialidade on Medico.CodEspec = Especialidade.CodEspec;
```

```
select Medico.NomeMed, Especialidade.NomeEspec
from Medico natural join Especialidade;
```

**E)** Especifique o comando SQL que retorne os nomes de todas as clínicas médicas e os respectivos nomes dos(as) médicos(as) vinculados(as) às clínicas. Considere apenas os(as) médicos cuja data de ingresso seja superior a 2015.

```
select Medico.NomeMed, Clinica.NomeCli
from ClinicaMedico join Medico using(CodMed) join Clinica using(CodCli)
where ClinicaMedico.DataIngresso >= '2016-01-01';
```

**F)** Especifique o comando SQL que retorne os códigos e nomes dos médicos que **não atuam** em nenhuma das clínicas cadastradas no banco de dados. Apresentar a consulta de três formas: (I) *left join*, (II) *not in*, (III) *not exists*. Apresente também a tabela resultante.

```
select Medico.CodMed, Medico.NomeMed
from Medico left outer join ClinicaMedico using(CodMed)
```

where ClinicaMedico.CodMed is null;

select distinct Medico.CodMed, Medico.NomeMed

from Medico, ClinicaMedico

where Medico.CodMed not in (select CodMed from ClinicaMedico);

select Medico.CodMed, Medico.NomeMed

from Medico

where not exists (select CodMed from ClinicaMedico where ClinicaMedico.CodMed = Medico.CodMed);

**G)** Especifique o comando SQL que retorne os nomes de todas as clínicas médicas e a quantidade de médicos(as) de cada clínica.

select ClinicaMedico.CodCli, count(ClinicaMedico.CodMed)  
from Medico right join ClinicaMedico using(CodMed)  
group by ClinicaMedico;

**H)** Especifique o comando SQL que retorne os nomes das clínicas médicas que estão sem médicos cadastrados.

select Clinica.NomeCli as "Clinica Sem Médico"

from Clinica left join ClinicaMedico using(CodCli)

where ClinicaMedico.CodMed is null;

**I)** Explique o retorno da consulta SQL abaixo:

```
SELECT Medico.CodMed
FROM Medico
WHERE
    NOT EXISTS (
        SELECT * FROM Clinica
        WHERE
            NOT EXISTS (
                SELECT * FROM ClinicaMedico
                WHERE Clinica.CodCli
                    = ClinicaMedico.CodCli AND
                    Medico.CodMed
                    = ClinicaMedico.CodMed
            )
    )
)
```

A consulta contém alguns erros que impedem a execução, falta um ponto e vírgula ( ; ) no final do código, ao ler o código se entende que é um trecho que deve retornar CodMed para qual não existe Clínica associado a ele

**J)** Especifique o comando SQL que retorne:

- Código e nome de cada especialidade;
- Código e nome de cada médico que atua nesta especialidade (Atenção: se a especialidade não tiver médicos, deve-se aparecer null);
- Código e nome de cada clínica em que atua o médico (Atenção: se o médico não atua em nenhuma clínica, deve-se aparecer null).

Atenção: na letra J deve-se apresentar uma única consulta SQL (único *select*), e não 3.

```
select Especialidade.CodEspec, Especialidade.NomeEspec, Medico.CodMed, Medico.NomeMed,
ClinicaMedico.CodCli, Clinica.NomeCli
from Especialidade left join Medico using(CodEspec)
left join ClinicaMedico using(CodMed) left join Clinica using (CodCli);
```

**K)** Elabore uma consulta (escolha livre) que represente a operação “*Full Outer Join*”. Lembrando que o MySQL não implementa diretamente esta operação, sendo necessário usar a operação de união. Explique o significado da consulta.

```
select Medico.NomeMed as NomeMed, RCN.NomeCli
from Medico left join (select * from ClinicaMedico left join Clinica using(CodCli)) as RCN using(CodMed)
union all
select Medico.NomeMed, RCNX.NomeCli
from Medico right join (select * from ClinicaMedico right join Clinica using(CodCli)) as RCNX using(CodMed)
where Medico.CodMed is null;
```

Select nome do médico e da clinica, saber quais medicos estao em clínica e quais nao estao

**L)** Especifique uma **visão (view)** em SQL que retorne para cada especialidade, o código e o nome da especialidade, seguido do número de médicos (quantidade total) que atuam na especialidade. Considere que apenas as especialidades com mais de 10 médicos devem aparecer no resultado da consulta.

**M)** Especifique um **gatilho (Trigger)** em SQL para o BD Clínicas Médicas (escolha livre). Explique o objetivo de seu gatilho e apresente um exemplo do gatilho sendo ativado/executado pelo SGBD.

```
delimiter //
create trigger update_ClinicaMedico before update
on ClinicaMedico
for each row -- para cada linha que for alterada, execute essas instruções:
begin
-- se durante um update não for fornecido uma carga horária, será adicionado o valor 20 horas
if (NEW.CargaHorariaSemanal = OLD.CargaHorariaSemanal) then
set NEW.CargaHorariaSemanal = 22;
end if;

end //

delimiter;
```

toda vez que fizer um update na ClinicaMedico e o valor de CargaHorariaSemanal for vazio, o valor será 22

**N)** Especifique um **procedimento armazenado (Stored Procedure)** em SQL para o BD Clínicas Médicas (escolha livre). Explique o objetivo de seu procedimento armazenado e apresente um exemplo dele sendo ativado/executado pelo SGBD.

DELIMITER //

```
CREATE PROCEDURE Quant_de_Medicos_Nao_Associados_a_Clinica (out total int)
BEGIN
```

```
    select count(CodMed) into total from Medico left join ClinicaMedico using(CodMed) where CodCli is null;
END //
DELIMITER ;
```

```
call Quant_de_Medicos_Nao_Associados_a_Clinica(@quant);
select @quant;
```

quantidade de medicos que nao estao associados a ClinicaMedico

**O)** Apresente a expressão da **álgebra relacional** E o comando **SQL** que retorna os nomes de todas as médicas (apenas gênero feminino) cuja especialidade é Pediatria.

**P)** Apresente a expressão da **álgebra relacional** E o comando **SQL** que retorna os nomes dos(as) médicos(as) dermatologias, cuja carga horária semanal é maior que 20 horas.

**Q)** Apresente a expressão da **álgebra relacional** E o comando **SQL** que retorna os nomes dos(as) pacientes que possuem consultas agendadas entre 12/08/2021 e 12/08/2022.

**R)** Apresente a expressão da **álgebra relacional** E o comando **SQL** que retorna os nomes dos médicos que iniciam com a letra 'A', suas respectivas especialidades e os nomes das clínicas em que atuam.

## QUESTÃO 2 (2,0 pontos)

Por meio da **Engenharia Reversa**, apresente a imagem do diagrama Entidade-Relacionamento Estendido (EER), gerado automaticamente por meio do banco de dados relacional Clínicas Médicas. Verificar e corrigir eventuais inconsistências nas cardinalidades (mínimas e máximas) do diagrama EER.

**Bons Estudos!**