



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

2ª ATIVIDADE

Banco de Dados Relacional MySQL na Prática

Estudante (nome completo): Estom Paulino da Silva Junior

Pontuação da atividade: 10 pontos

Prazo: 02/08/2022

Conteúdos: Capítulos 3 e 4 do livro “Sistemas de Banco de Dados”, 6ª Edição, Autores: Elmasri e Navathe.

⚠ **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 (5,0 PONTOS)

Considere o estado/instância do banco de dados Startup apresentado abaixo (as PK estão sublinhadas):

Startup			Linguagem_Programação		
<u>id_startup</u>	nome_startup	cidade_sede	<u>id_linguagem</u>	nome_linguagem	ano_lançamento
10001	Tech4Toy	Porto Alegre	20001	Python	1991
10002	Smart123	Belo Horizonte	20002	PHP	1995
10003	knowledgeUp	Rio de Janeiro	20003	Java	1995
10004	BSI Next Level	Recife	20004	C	1972
10005	QualiHealth	São Paulo	20005	JavaScript	1995
10006	ProEdu	Florianópolis	20006	Dart	2011

Programador					
<u>id_programador</u>	id_startup	nome_programador	gênero	data_nascimento	email
30001	10001	João Pedro	M	23/06/1993	joaop@mail.com
30002	10002	Paula Silva	F	10/01/1986	paulas@mail.com
30003	10003	Renata Vieira	F	05/07/1991	renatav@mail.com
30004	10004	Felipe Santos	M	25/11/1976	felipes@mail.com
30005	10001	Ana Cristina	F	19/02/1968	anac@mail.com
30006	10004	Alexandre Alves	M	07/07/1988	alexandrea@mail.com
30007	10002	Laura Marques	F	04/10/1987	lauram@mail.com

Programador_Linguagem	
<u>id_programador</u>	<u>id_linguagem</u>
30001	20001
30001	20002
30002	20003
30003	20004
30003	20005
30004	20005
30007	20001
30007	20002

Figura 1 - Estado de um BD que armazena informações de startups, linguagens e seus programadores.

A) Apresente o link (*google drive* ou *github*) contendo o **script SQL** com as instruções/comandos de criação das 5 tabelas do esquema Startup, sendo as 4 tabelas ilustradas na Figura 1, além da tabela extra que você idealizou na Atividade 1.

Ao desenvolver o script SQL de criação e carga (inserção dos dados) do BD Startup lembre-se de especificar todas as chaves primárias e estrangeiras das tabelas, assim como incluir as seguintes restrições de integridade (itens I a VII):

- I. O atributo *ano_lançamento* é do tipo/domínio YEAR e os atributos *id_startup*, *id_linguagem* e *id_programador* são do tipo CHAR(5). Os tipos dos demais atributos podem ser escolhidos livremente.
- II. Os atributos *nome_startup*, *nome_programador* e *nome_linguagem* não podem ser nulos.
- III. O e-mail do(a) programador(a) tem que ser único, ou seja, não podem ter dois ou mais programadores(as) com o mesmo e-mail cadastrado.
- IV. Nenhuma linguagem de programação pode ser excluída caso exista algum(a) programador(a) associado(a) a ela.
- V. Ao excluir um (ou mais) programador(a), as informações sobre as linguagens em que ele(ela) programa devem ser excluídas automaticamente.
- VI. Ao alterar o código de uma startup, todos(as) os(as) programadores(as) que estão associados(as) a este código devem ter seus códigos de startup atualizados automaticamente.
- VII. Elabore uma restrição de integridade referencial, considerando a tabela que você criou.

Dica: para implementar as restrições IV a VII devem ser utilizadas as ações de disparo referencial '*on delete*' e/ou '*on update*'.

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

B) Com base no estado do BD Startup da Figura 1, **apresente exemplos de comandos SQL (podendo ser inserção, remoção ou atualização) que violem as cinco restrições de integridade elencadas abaixo**, e dessa forma constate que as referidas restrições estão sendo aplicadas/executadas corretamente pelo SGBD. (Atenção: deverão constar 5 comandos SQL, isto é, um comando para cada restrição.)

1. Restrição de Integridade de chave (singularidade) → Exemplificar considerando o atributo *id_linguagem*.

```
insert into LINGUAGEM_PROGRAMACAO
```

```
values ('20008','Python','1991'),
```

```
      ('20008','Java','1995')
```

constatação de violação de integridade de chave, Error Code: 1062. Duplicate entry '20008' for key 'PRIMARY'

2. Restrição de Integridade de domínio → Exemplificar considerando o atributo *ano_lançamento*.

```
insert into LINGUAGEM_PROGRAMACAO
```

```
values ('20001','Python','ano 1991')
```

Error Code

3. Restrição de integridade de entidade → Exemplificar considerando o atributo *id_programador*

```
insert into Programador values
```

```
(null,'30001','João Pedro', 'M','1993-06-23', 'joaop@mail.com')
```

constatação da violação: Error Code: 1048. Column 'id_programador' cannot be null

4. Restrição de integridade de vazio (null) → Exemplificar considerando o atributo *nome_programador*.

```
insert into PROGRAMADOR
```

```
value (null, 10008, 'João Pedro', 'M', 1993-06-23, 'joaop@mail.com');
```

- constatação da violação: Error Code: 1048. Column 'nome_programador' cannot be null

5. Restrição de integridade referencial → Exemplificar considerando o atributo *id_startup*.

`insert into PROJETOS_STARTUP values`

`('50001','Apollo', '10022', '20003')`

(observação: a restrição de integridade semântica (regras de negócio) só será abordada na atividade 3, com o uso de gatilhos (*trigger*) e procedimentos armazenados (*stored procedures*)).

ATENÇÃO: Todas as consultas que envolverem JUNÇÕES devem ser feitas por meio da cláusula WHERE. Não utilizar INNER JOIN, NATURAL JOIN, LEFT JOIN e afins, visto que tais comandos serão abordados a partir da 5ª semana, por meio da 3ª atividade.

C) Apresente a consulta SQL que retorna as informações da tabela abaixo:

	nome_programador	nome_startup
▶	João Pedro	Tech4Toy
	Paula Silva	Smart123
	Renata Vieira	knowledgeUp
	Felipe Santos	BSI Next Level
	Ana Cristina	Tech4Toy
	Alexandre Alves	BSI Next Level
	Laura Marques	Smart123

```
select p.nome_programador, s.nome_startup
from PROGRAMADOR as p, STARTUP as s
where p.id_startup = s.id_startup;
```

D) Apresente a consulta SQL que liste os nomes dos(as) programadores(as) que programam em Java ou C.

```
select P.nome_programador
from PROGRAMADOR as P, PROGRAMADOR_LINGUAGEM as PL, LINGUAGEM_PROGRAMACAO as LP
where P.id_programador = PL.id_programador AND LP.id_linguagem = PL.id_linguagem
AND (LP.nome_linguagem = 'Java' or LP.nome_linguagem = 'C');
```

E) Apresente a consulta SQL que lista o nome de cada programador(a), a data de nascimento no **formato dd/mm/aaaa**, a data de nascimento **por extenso** e a **idade atual**. A tabela resultante deve ser similar a que segue abaixo:

nome_programador	dd/mm/aaaa	Data_por_Extenso	Idade
João Pedro	23/06/1993	23 de junho de 1993	29
Paula Silva	10/01/1986	10 de janeiro de 1986	36
Renata Vieira	05/07/1991	05 de julho de 1991	31
Felipe Santos	25/11/1976	25 de novembro de 1976	45
Ana Cristina	19/02/1968	19 de fevereiro de 1968	54
Alexandre Alves	07/07/1988	07 de julho de 1988	34
Laura Marques	04/10/1987	04 de outubro de 1987	34

Observação: o idioma de exibição dos meses (por extenso) no MySQL é inglês por padrão. Por isso, faz-se necessário configurar a variável de sistema `lc_time_names` para que os meses sejam apresentados no idioma português. Segue comando:

```
select nome_programador,  
date_format(data_nascimento, '%d/%m/%Y') AS data_nascimento,  
date_format(data_nascimento, '%d de %M de %Y') AS data_extenso,  
timestampdiff(year, data_nascimento, curdate()) as Idade  
from PROGRAMADOR
```

F) Apresente a consulta SQL que lista os nomes dos(as) programadores(as), suas idades, e os nomes das startups em que estão vinculados(as), mas APENAS os programadores(as) que estejam vinculados(as) às startups “Smart123” ou “BSI Next Level” e tenham atualmente entre 30 e 40 anos. A tabela resultante referente a esta consulta segue abaixo:

nome_programador	Idade	nome_startup
Paula Silva	36	Smart123
Laura Marques	34	Smart123
Alexandre Alves	34	BSI Next Level

```
select nome_programador,  
timestampdiff(year, data_nascimento, curdate()) as Idade,  
nome_startup  
from PROGRAMADOR as P, STARTUP as S  
where p.id_startup = s.id_startup  
AND (nome_startup = "Smart123" or nome_startup = "BSI Next Level")  
AND timestampdiff(year, data_nascimento, curdate()) between 30 and 40
```

QUESTÃO 2 (1,5 PONTOS)

Semelhante ao que feito na letra A da 1ª questão (incluindo as restrições de integridade), apresente o link (google drive ou github) contendo o **script SQL (NOVA VERSÃO)** com as instruções/comandos de criação das 5 tabelas do esquema Startup, sendo as 4 tabelas ilustradas na Figura 1, além da tabela extra que você idealizou na Atividade 1, CONTUDO, as **chaves primárias** das tabelas *Startup*, *Linguagem_Programação* e *Programador* devem numéricas nesta nova versão do *script* (neste caso, do tipo INT) e **AUTOINCREMENTO**. No caso da tabela *Startup* o *autoincremento* da PK deve *iniciar* em 10001; na tabela *Linguagem_Programação* o *autoincremento* da PK deve *iniciar* em 20001; e na tabela *Programador* o *autoincremento* da PK deve *iniciar* em 30001.

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

QUESTÃO 3 (3,5 PONTOS)

Considere o banco de dados relacional Empresa, cujo *script* (arquivo “Script_Empresa.sql”) para criação e carga do referido banco encontra-se disponível no Classroom, no Guia de Estudos das Semanas 03 e 04. A figura 3.6 (página 48) do livro “Sistemas de Banco de Dados - 6ª Edição” ilustra o estado do BD Empresa.

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

A) Apresente o comando SQL que retorna os nomes e localizações (cidades) de todos os projetos vinculados ao departamento Pesquisa, cujas localizações iniciam com a letra 'S'. A tabela resultante referente a esta consulta segue abaixo:

Projnome	Projlocal
ProdutoX	Santo André
ProdutoZ	São Paulo

```
select Projnome,Projlocal
from PROJETO where Projlocal LIKE 's%'
```

B) Insira na tabela "projeto" as informações abaixo (apresente o comando SQL de inserção), em seguida execute novamente o comando da letra A e apresente a tabela resultante.

Projnome: ProjetoBSI	Projnumero: 40	Projlocal: Salvador	Dnum: 5
-------------------------	-------------------	------------------------	------------

```
insert into PROJETO values
('ProjtoBSI',40, 'Salvador', 5)
```

Result Grid		Filter Rows:
	Projnome	Projlocal
▶	ProdutoX	Santo André
	ProdutoZ	São Paulo
	Reorganização	São Paulo
	ProjtoBSI	Salvador

C) Apresente o comando SQL que retorna os nomes de todos(as) funcionários(as) que são supervisionados diretamente por Fernando. Apresente também a tabela resultante.

```
from FUNCIONARIO
where FUNCIONARIO.Cpf_supervisor = (
    select FUNC.Cpf
    from FUNCIONARIO as FUNC
    where FUNC.Pnome = 'Fernando' and FUNC.Unome = 'Wong');
```

Result Grid		Filter Rows:
	Pnome	Unome
▶	João	Silva
	Joice	Leite
	Ronaldo	Lima

D) Apresente o comando SQL que retorna os nomes dos departamentos e os nomes dos gerentes, mas apenas os departamentos gerenciados por homens. Apresente também a tabela resultante.

```
select P.Pnome, Dnome
from FUNCIONARIO as P, DEPARTAMENTO as PL
Where P.dnr = PL.Dnumero
AND (P.Sexo = 'M');
```

Result Grid			Filter Rows:
	Pnome	Dnome	
▶	João	Pesquisa	
	Fernando	Pesquisa	
	Jorge	Matriz	
	André	Administração	

E) Apresente o comando SQL que retorna os nomes dos(as) funcionários(as) e respectivos nomes dos projetos em que eles(as) atuam. Considerar apenas os projetos vinculados ao departamento cuja gerente é uma mulher, além disso, considerar apenas os(as) funcionários(as) que trabalham mais de 25 horas por semana. A tabela resultante referente a esta consulta segue abaixo:

pnome	Projnome
André	Informatização
Alice	NovosBenefícios

F) Modificar a gerência do departamento Administração de Jennifer para Alice, bem como a data de início da gerência para 01/07/2022. Em seguida, já que Alice foi promovida a gerente, criar outro comando que modifique o seu salário, adicionando 80% (novo salário de Alice: 45.000).

G) Explique as violações de integridade que ocorrem ao executar os comandos abaixo:

- Modificar o supervisor do funcionário Fernando para o cpf 55689742328 (apresente o comando SQL de atualização).

UPDATE FUNCIONARIO

SET Cpf_supervisor ='55689742328'

WHERE Cpf = '33344555587';

integridade referencial

- Deletar as informações do funcionário Ronaldo da tabela funcionário (apresente o comando SQL de remoção).

DELETE FROM FUNCIONARIO

WHERE Cpf = '66688444476';

integridade referencial

H) Explique por que NÃO ocorrem violações de integridade ao executar os comandos abaixo:

- Deletar as informações do funcionário Ronaldo da tabela “trabalha_em” (apresente o comando SQL de remoção).

DELETE FROM TRABALHA_EM

WHERE Fcpf = '66688444476';

- Após remoção das informações do funcionário Ronaldo da tabela “trabalha_em”, deletar novamente as informações do funcionário Ronaldo da tabela “funcionario”.

```
1 • select * from FUNCIONARIO
2 WHERE Cpf = '66688444476'
```

Result Grid										
		Filter Rows:		Edit:		Export/Import:		Wrap Cell Content:		
	Pnome	Minicial	Unome	Cpf	Datanasc	Endereco	Sexo	Salario	Cpf_supervisor	Dnr
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

I) Dê exemplos de chaves primárias naturais e artificiais considerando o BD Empresa, e seguida, descreva as vantagens e desvantagens do uso de cada uma delas.

Bons estudos!

Em caso de dúvidas, entrar em contato com a profa. Roberta ou a monitora Aurilene.