

Design e Desenvolvimento de Bancos de Dados

— PSet 2 —

Prof. Abrantes Araújo Silva Filho

Data de Entrega: 12/05/2022

LEIA COM ATENÇÃO!

Um *Problem Set* (PSet) é um conjunto de problemas e tarefas difíceis (alguns extremamente difíceis) que o forçarão a estudar e realmente compreender a matéria¹. É **essencial** que você inicie o PSet o mais rápido possível... deixar para começar um PSet faltando um ou dois dias da data de entrega significa que você não conseguirá terminar. Também é **fundamental** que você, em caso de dúvidas ou dificuldades, participe das monitorias e discuta suas dúvidas com o monitor e/ou professor.

Sumário

1	Instruções	2
1.1	Dependência do PSet-1	3
1.2	Como entregar este PSet?	3
2	Relatórios SQL a serem desenvolvidos	4

¹Para maiores informações sobre os PSets, leia a Seção 5.3 do *Syllabus* da disciplina.

1 Instruções

Este PSet é uma das atividades **pontuadas** que, conforme detalhado no *Syllabus* (Seções 5 e 6), terão peso de 45% na nota final da disciplina. Por favor siga todas as instruções abaixo para a realização e entrega deste PSet.

Este PSet tem um grande objetivo:

- Fazer com que você aprenda a *Structured Query Language* (SQL) em nível básico e intermediário.

O PSet pode ser feito em grupos de até três alunos desde que as seguintes regras sejam absolutamente seguidas:

1. **Respostas individuais:** o grupo pode discutir o assunto e tirar dúvidas entre si mas, ao final, cada aluno deve escrever sua resposta de forma **separada, individual e distinta** das respostas de seus colegas;
2. **Trabalho integral:** todos os participantes de grupo devem trabalhar de forma integral em **todas as questões**. É uma infração às normas de Integridade Acadêmica desta disciplina dividir questões, por exemplo: um aluno responde um grupo de questões, outro aluno responde outro grupo de questões, e um outro aluno responde ao resto das questões e, depois, todos copiam as respostas entre si. Isso não é permitido, cada aluno deve resolver todas as questões;
3. **Código individual:** cada aluno deve escrever e preparar seu próprio código SQL (incluindo scripts ou o que mais for necessário). Não é permitido compartilhar código entre os alunos; e
4. **Identifique os colegas:** se você trabalhar em grupo você deve identificar, em todos os documentos, códigos, scripts, etc., o nome dos colegas com os quais você trabalhou.

Atenção para as regras de trabalho colaborativo!

Por favor, **LEIA ATENTAMENTE** as regras sobre Integridade Acadêmica do *Syllabus* (Seção 7), em especial a “**Política sobre trabalho colaborativo**” (Seção 7.1), para você saber **como trabalhar em grupo** nesta disciplina. Atividades que quebrarem essas regras terão a nota zerada e os alunos encaminhados à coordenação da UVV para aplicação das penalidades previstas (exceto quando o aluno utilizar a “**Cláusula de Arrependimento**”, Seção 7.2 do *Syllabus*).

As atividades deste PSet devem ser realizadas, preferencialmente, na **Máquina Virtual** da disciplina, que já tem o ambiente todo configurado, os softwares instalados e está pronta para uso. A máquina virtual pode ser baixada no site do Computação Raiz (<https://www.computacaoraiz.com.br>). Caso você opte por não utilizar

a máquina virtual, terá de instalar todos os SGBD em seu próprio computador e configurar o ambiente por conta própria (é difícil, mas não impossível).

Antes de começar, leia integralmente o PSet para ter uma noção geral do que será exigido, para estimar o grau de dificuldade que você terá, e para estimar quantas horas por dia você precisará se dedicar ao PSet. Em caso de dúvidas, entre em contato com o professor ou com os monitores o mais rápido possível.

Lembre-se: deixar para começar um PSet faltando apenas um ou dois dias da data de entrega significa que você não conseguirá terminar! Comece cedo, trabalhe um pouco todo dia, mantenha-se atento à data de entrega e tire suas dúvidas com os monitores e/ou professor. Não ultrapasse a data de entrega!

1.1 Dependência do PSet-1

Este PSet-2 depende totalmente do PSet-1² pois, agora, você deverá gerar diversos relatórios SQL a partir dos dados que você inseriu no banco de dados criado no primeiro PSet.

ATENÇÃO: Se você não fez o PSet-1, não conseguirá fazer o PSet-2 e, nesse caso, você precisará trabalhar em dobro: fazer o PSet-1 e o PSet2. Peça ajuda aos monitores sempre que precisar.

1.2 Como entregar este PSet?

A entrega do resultado deste PSet será feita exclusivamente através do repositório padronizado que você criou no GitHub, durante o PSet-1. Faça o seguinte:

1. Dentro do repositório da disciplina (que foi criado com o nome padronizado `uvv_bd_1_TURMA`), crie o subdiretório `pset2`;
2. Faça a documentação do diretório `pset2` utilizando Markdown;
3. Dentro do diretório deste PSet crie um script SQL com as respostas às questões da próxima seção;
4. Envie um e-mail para o professor (`abrantesasf@uvv.br`), até a data limite de entrega, especificando: seu nome completo, matrícula, turma e URL do repositório GitHub;
5. Observação: mesmo que você inclua ou modifique arquivos no repositório GitHub após a data limite, somente serão avaliadas as respostas comitadas no repositório até a data limite.

²Durante o PSet-1 você criou o seu usuário, o banco de dados, os esquemas, as tabelas, os relacionamentos e, por fim, fez a inserção dos dados em todas as tabelas.

2 Relatórios SQL a serem desenvolvidos

QUESTÃO 01: prepare um relatório que mostre a média salarial dos funcionários de cada departamento.

```
select numero_departamento, round (avg(salario),2) as média_salarial from funcionario group by
numero_departamento order by numero_departamento;
```

QUESTÃO 02: prepare um relatório que mostre a média salarial dos homens e das mulheres. `select sexo, round (avg(salario),2) as média_salarial from funcionario group by sexo;`

QUESTÃO 03: prepare um relatório que liste o nome dos departamentos e, para cada departamento, inclua as seguintes informações de seus funcionários: o nome completo, a data de nascimento, a idade em anos completos e o salário.

```
select concat(primeiro_nome, nome_meio, ultimo_nome) as nome_completo, data_nascimento, year(current_timestamp())-year(data_nascimento) as idade,
salario, d.nome_departamento from funcionario f inner join departamento d on f.numero_departamento = d.numero_departamento ;
```

QUESTÃO 04: prepare um relatório que mostre o nome completo dos funcionários, a idade em anos completos, o salário atual e o salário com um reajuste que obedece ao seguinte critério: se o salário atual do funcionário é inferior a 35.000 o reajuste deve ser de 20%, e se o salário atual do funcionário for igual ou superior a 35.000 o reajuste deve ser de 15%.

```
select distinct concat
(f1.primeiro_nome,
f1.nome_meio,
f1.ultimo_nome) as
nome_completo,
year
(current_timestamp())-year
(f1.data_nascimento) as
idade,
f1.salario as
salario_atual,
if(f2.salario >=
35000, f2.salario * 1.15,
f2.salario * 1.20) as
salario_reajustado
from funcionario f1
inner join funcionario f2 on
(f1.cpf = f2.cpf);
```

QUESTÃO 05: prepare um relatório que liste, para cada departamento, o nome do gerente e o nome dos funcionários. Ordene esse relatório por nome do departamento (em ordem crescente) e por salário dos funcionários (em ordem decrescente). `select distinct f1.primeiro_nome as funcionarios, f2.primeiro_nome as gerentes, d.nome_departamento, f1.salario`

```
from funcionario f1
inner join funcionario f2 on (f1.cpf_supervisor = f2.cpf)
inner join departamento d on (f1.numero_departamento = d.numero_departamento)
order by nome_departamento, salario desc;
```

QUESTÃO 06: prepare um relatório que mostre o nome completo dos funcionários que têm dependentes, o departamento onde eles trabalham e, para cada funcionário, também liste o nome completo dos dependentes, a idade em anos de cada dependente e o sexo (o sexo NÃO DEVE aparecer como M ou F, deve aparecer como “Masculino” ou “Feminino”).

```
select distinct concat(primeiro_nome, nome_meio, ultimo_nome) as nome_funcionario, numero_departamento, concat(d.nome_dependente, f.nome_meio,
f.ultimo_nome) as nome_dependente, year(current_timestamp())-year(d.data_nascimento) as idade_dependente, if(d.sexo = "M","Masculino","Feminino") as sexo from
funcionario f, dependente
inner join dependente d where f.cpf = d.cpf_funcionario;
```

QUESTÃO 07: prepare um relatório que mostre, para cada funcionário que NÃO TEM dependente, seu nome completo, departamento e salário.

```
select concat(f.primeiro_nome, f.nome_meio, f.ultimo_nome) as nome_completo, f.salario, part.nome_departamento from funcionario as f inner join departamento part
ON (part.numero_departamento = f.numero_departamento) left join dependente depart ON (depart.cpf_funcionario = f.cpf) WHERE depart.nome_dependente IS null;
```

QUESTÃO 08: prepare um relatório que mostre, para cada departamento, os projetos desse departamento e o nome completo dos funcionários que estão alocados em cada projeto. Além disso inclua o número de horas trabalhadas por cada funcionário, em cada projeto.

```
select distinct numero_departamento, trabalha_em.numero_projeto, concat(f.primeiro_nome, f.nome_meio, f.ultimo_nome) as nome_funcionario, horas
from funcionario as f
inner join trabalha_em on f.cpf = trabalha_em.cpf_funcionario order by numero_departamento;
```

QUESTÃO 09: prepare um relatório que mostre a soma total das horas de cada projeto em cada departamento. Obs.: o relatório deve exibir o nome do departamento, o nome do projeto e a soma total das horas.

```
select sum(t.horas) as horas_totais, nome_projeto, nome_departamento from trabalha_em t inner join projeto p on p.numero_projeto = t.numero_projeto inner join departamento d on d.numero_departamento = p.numero_departamento where t.numero_projeto = t.numero_projeto group by t.numero_projeto, p.nome_projeto, d.nome_departamento;
```

QUESTÃO 10: prepare um relatório que mostre a média salarial dos funcionários de cada departamento.

```
select numero_departamento, round (avg(salario),2) as média_salarial from funcionario group by numero_departamento order by numero_departamento;
```

QUESTÃO 11: considerando que o valor pago por hora trabalhada em um projeto é de 50 reais, prepare um relatório que mostre o nome completo do funcionário, o nome do projeto e o valor total que o funcionário receberá referente às horas trabalhadas naquele projeto.

```
select concat(f.primeiro_nome,' ', nome_meio,' ',ultimo_nome) as nome_funcionario, projeto.nome_projeto,(trabalha_em.horas * 50) as valor from ((funcionario as f inner join trabalha_em on f.cpf = trabalha_em.cpf_funcionario) inner join projeto on projeto.numero_projeto = trabalha_em.numero_projeto);
```

QUESTÃO 12: seu chefe está verificando as horas trabalhadas pelos funcionários nos projetos e percebeu que alguns funcionários, mesmo estando alocados à algum projeto, não registraram nenhuma hora trabalhada. Sua tarefa é preparar um relatório que liste o nome do departamento, o nome do projeto e o nome dos funcionários que, mesmo estando alocados a algum projeto, não registraram nenhuma hora trabalhada.

```
select concat(primeiro_nome, nome_meio, ultimo_nome) as funcionario, d.nome_departamento, p.nome_projeto, horas from departamento d inner join projeto p on (d.numero_departamento = p.numero_departamento) inner join trabalha_em t on (t.numero_projeto = p.numero_projeto) inner join funcionario f on (t.cpf_funcionario = f.cpf) where t.horas = 0;
```

```
13
select distinct concat
(primeiro_nome,
nome_meio,
ultimo_nome) as
nome, sexo, year
(current_timestamp
))-year
(data_nascimento)
as idade
from funcionario
union
select distinct concat
(d.nome_dependente
e, f.nome_meio,
f.ultimo_nome) as
nome, d.sexo, year
(current_timestamp
))-year
(d.data_nascimento)
as idade
from dependente d
inner join
funcionario f on
(f.cpf =
d.cpf_funcionario)
order by idade desc;
```

QUESTÃO 13: durante o natal deste ano a empresa irá presentear todos os funcionários e todos os dependentes (sim, a empresa vai dar um presente para cada funcionário e um presente para cada dependente de cada funcionário) e pediu para que você preparasse um relatório que listasse o nome completo das pessoas a serem presenteadas (funcionários e dependentes), o sexo e a idade em anos completos (para poder comprar um presente adequado). Esse relatório deve estar ordenado pela idade em anos completos, de forma decrescente.

QUESTÃO 14: prepare um relatório que exiba quantos funcionários cada departamento tem.

```
select f.numero_departamento, count(*) as qnt_empregados from funcionario f, departamento d
where f.numero_departamento = d.numero_departamento
group by f.numero_departamento;
```

QUESTÃO 15: como um funcionário pode estar alocado em mais de um projeto, prepare um relatório que exiba o nome completo do funcionário, o departamento desse funcionário e o nome dos projetos em que cada funcionário está alocado. Atenção: se houver algum funcionário que não está alocado em nenhum projeto, o nome completo e o departamento também devem aparecer no relatório.

```
select distinct concat(f.primeiro_nome, f.nome_meio, f.ultimo_nome) as nome_completo, f.numero_departamento, p.nome_projeto
from funcionario f
inner join trabalha_em te on (f.cpf = te.cpf_funcionario)
left outer join projeto p on (p.numero_projeto = te.numero_projeto and f.cpf = te.cpf_funcionario and te.horas > 0)
order by nome_completo;
```