



Curso: Licenciatura em Engenharia Informática (LEI);
Mestrado Integrado em Engenharia Informática (MIEI)
Unidade Curricular: Bases de Dados
Ano Letivo: 2021/2022 – 2º Semestre

FICHA DE EXERCÍCIOS - PL08

Desenvolva as instruções SQL necessárias de forma a se poder responder às seguintes questões:

Questão 1. Qual é o nome dos medicamentos que não terminam em 'ol' nem em 'na'?

```
SELECT nome FROM farmacos WHERE nome NOT LIKE '%ol' AND nome NOT LIKE '%na';  
  
SELECT nome FROM farmacos WHERE nome NOT RLIKE 'ol$' AND nome NOT RLIKE 'na$';  
  
SELECT nome FROM farmacos WHERE nome NOT REGEXP 'ol$' AND nome NOT REGEXP  
'na$';
```

Questão 2. Retorne a morada completa dos pacientes com mais de 50 anos.

```
SELECT concat(rua, ' ', localidade, ' ', cod_postal) as morada_completa  
FROM pacientes WHERE TIMESTAMPDIFF(YEAR, dta_nascimento, CURDATE()) > 50;
```

-- apenas se já tiver criado a função idade

```
SELECT concat(rua, ' ', localidade, ' ', cod_postal) as morada_completa  
FROM pacientes WHERE idade(dta_nascimento) > 50;
```

Questão 3. Liste as datas de prescrição e de validade de todas as prescrições da BD no formato '%d/%m/%Y' bem como o número de dias em que estas são válidas para serem levantadas numa farmácia.

-- Como o enunciado desta questão não estava 100% claro dava margem para dois tipos de resposta:

-- número de dias de validade de cada prescrição

```
SELECT DATE_FORMAT(data_validade, '%Y-%m-%d') data_validade,  
DATE_FORMAT(data_prescricao, '%Y-%m-%d') data_prescricao,  
datediff(data_validade, data_prescricao) as nr_dias_valido FROM prescricoes;
```

-- número de dias em que as prescrições ainda são válidas para serem levantadas numa farmácia

```
SELECT DATE_FORMAT(data_validade, '%Y-%m-%d') data_validade,  
DATE_FORMAT(data_prescricao, '%Y-%m-%d') data_prescricao,  
greatest(datediff(data_validade, CURDATE()), 0) as nr_dias_valido FROM prescricoes;
```

Questão 4. Liste as localidades dos pacientes que possuem exatamente 5 caracteres.

```
SELECT DISTINCT localidade FROM pacientes WHERE localidade LIKE '_____';  
SELECT localidade FROM pacientes WHERE localidade LIKE '_____ ' GROUP BY localidade;  
SELECT DISTINCT localidade FROM pacientes WHERE localidade REGEXP '^.{5}$';  
SELECT localidade FROM pacientes WHERE localidade REGEXP '^.{5}$' GROUP BY localidade;
```

Questão 5. Retorne para cada estado de agendamento (0 – livre e 1 – ocupado) o número total de agendamentos.

```
SELECT estado, count(*) as total FROM horario_agendamento GROUP BY estado;
```

Questão 6. Qual é a quantidade total de cada fármaco prescrito até ao momento?

```
SELECT f.nome, SUM(p.quantidade) as quantidade_total FROM prescricoes p, farmacos f  
WHERE p.id_farmaco = f.id_farmaco GROUP BY f.nome;  
SELECT f.nome, SUM(p.quantidade) as quantidade_total FROM farmacos f NATURAL JOIN  
prescricoes p GROUP BY f.nome;
```

Questão 7. Qual é a quantidade mínima de cada fármaco prescrito até ao momento?

```
SELECT f.nome, MIN(p.quantidade) as quantidade_total FROM prescricoes p, farmacos f  
WHERE p.id_farmaco = f.id_farmaco GROUP BY f.nome;  
SELECT f.nome, MIN(p.quantidade) as quantidade_total FROM farmacos f NATURAL JOIN  
prescricoes p GROUP BY f.nome;
```

Questão 8. Qual é o valor máximo de consulta cobrado por especialidade sem casas decimais?

-- Arredondando para cima (FORMAT(v, n), ROUND(v, n), CEIL(v))

-- Arredondando para baixo (TRUNCATE(v,n), FLOOR(v))

```
SELECT e.des_especialidade, FORMAT(MAX(c.preco), 0) as preco_max  
FROM consultas c, especialidades e, medicos m  
WHERE m.num_mec = c.id_medico AND m.cod_especialidade = e.cod_especialidade  
GROUP BY e.des_especialidade;  
SELECT e.des_especialidade, FORMAT(MAX(c.preco),0) as preco_max FROM especialidades e  
NATURAL JOIN medicos m INNER JOIN consultas c ON m.num_mec=c.id_medico GROUP BY  
e.des_especialidade;
```

Questão 9. Indique o número de pacientes para os sexos feminino e masculino.

```
SELECT sexo, COUNT(*) as nr_pacientes FROM pacientes GROUP BY sexo HAVING sexo IN ('F', 'M');
```

```
SELECT sexo, COUNT(*) as total_pacientes FROM pacientes WHERE sexo IN ('F', 'M') GROUP BY sexo;
```

Questão 10. Liste o número total mensal de consultas por ano que não foram faturadas, a partir de 2019, do ano mais recente para o mais antigo.

```
SELECT MONTH(hora_ini) as mes, YEAR(hora_ini) as ano, COUNT(*) as total FROM consultas WHERE id_secretaria IS NULL GROUP BY mes, ano HAVING ano>2019 ORDER BY ano DESC;
```

Questão 11. Liste os procedimentos que possuem um valor acima da média por ordem decrescente de preço.

```
SELECT * FROM procedimentos WHERE preco > (SELECT AVG(preco) FROM procedimentos) ORDER BY preco DESC;
```

Questão 12. Quais são os médicos de Oftalmologia que não consultaram pacientes em Janeiro de 2020?

```
SELECT m.* FROM medicos m, especialidades e WHERE e.des_especialidade = 'Oftalmologia' AND e.cod_especialidade = m.cod_especialidade AND m.num_mec NOT IN (SELECT m.num_mec FROM medicos m, consultas c WHERE MONTH(c.hora_ini)=1 AND YEAR(c.hora_ini)='2020' AND m.num_mec = c.id_medico);
```

```
SELECT m.* FROM medicos m NATURAL JOIN especialidades e WHERE e.des_especialidade = 'Oftalmologia' AND m.num_mec NOT IN (SELECT id_medico FROM consultas WHERE MONTH(hora_ini)=1 AND YEAR(hora_ini)=2020);
```

```
SELECT m.* FROM medicos m NATURAL JOIN especialidades e WHERE e.des_especialidade = 'Oftalmologia' AND NOT EXISTS (SELECT c.id_medico FROM consultas c WHERE MONTH(c.hora_ini)=1 AND YEAR(c.hora_ini)=2020 AND c.id_medico = m.num_mec);
```

Questão 13. Quais são as especialidades que não foram consultadas durante o mês de Janeiro de 2020?

```
SELECT e.* FROM medicos m, especialidades e WHERE e.cod_especialidade = m.cod_especialidade AND m.num_mec NOT IN (SELECT c.id_medico FROM consultas c WHERE MONTH(c.hora_ini)=1 AND YEAR(c.hora_ini)='2020') GROUP BY e.cod_especialidade;
```

```
SELECT DISTINCT e.* FROM especialidades e NATURAL JOIN medicos m WHERE m.num_mec NOT IN (SELECT id_medico FROM consultas WHERE MONTH(hora_ini)=1 AND YEAR(hora_ini)=2020);
```

```
SELECT e.* FROM especialidades e NATURAL JOIN medicos m WHERE m.num_mec NOT IN (SELECT id_medico FROM consultas WHERE MONTH(hora_ini)=1 AND YEAR(hora_ini)=2020) GROUP BY e.cod_especialidade;
```

```
SELECT DISTINCT e.* FROM especialidades e NATURAL JOIN medicos m WHERE NOT EXISTS (SELECT c.id_medico FROM consultas c WHERE MONTH(c.hora_ini)=1 AND YEAR(c.hora_ini)=2020 AND c.id_medico=m.num_mec);
```

```
SELECT e.* FROM especialidades e NATURAL JOIN medicos m WHERE NOT EXISTS (SELECT c.id_medico FROM consultas c WHERE MONTH(c.hora_ini)=1 AND YEAR(c.hora_ini)=2020 AND c.id_medico=m.num_mec) GROUP BY e.cod_especialidade;
```

Questão 14. Quais são os médicos que nunca consultaram pacientes de Braga?

```
SELECT * FROM medicos WHERE num_mec NOT IN (SELECT c.id_medico FROM consultas c, pacientes p WHERE p.localidade = 'Braga' AND p.nr_sequencial = c.id_paciente);
```

```
SELECT * FROM medicos WHERE num_mec NOT IN (SELECT c.id_medico FROM consultas c INNER JOIN pacientes p ON c.id_paciente = p.nr_sequencial WHERE p.localidade = 'Braga' );
```

```
SELECT * FROM medicos m WHERE NOT EXISTS (SELECT c.id_medico FROM consultas c INNER JOIN pacientes p ON c.id_paciente = p.nr_sequencial WHERE p.localidade = 'Braga' AND m.num_mec = c.id_medico);
```

Questão 15. Quantos dias passaram desde que o paciente 111113 foi consultado pela última vez?

```
SELECT DATEDIFF(CURDATE(), hora_ini) FROM consultas WHERE hora_ini = (SELECT MAX(hora_ini) FROM consultas WHERE id_paciente=111113);
```

```
SELECT DATEDIFF(CURDATE(), hora_ini) FROM consultas WHERE hora_ini IN (SELECT MAX(hora_ini) FROM consultas WHERE id_paciente=111113);
```

Questão 16. Quais são os pacientes que só foram consultados em Obstetrícia?

-- Pacientes que foram consultados a obstetrícia – os pacientes que foram consultados a outras especialidades

```
SELECT p.* FROM pacientes p INNER JOIN consultas c ON p.nr_sequencial=c.id_paciente INNER JOIN medicos m ON c.id_medico=m.num_mec NATURAL JOIN especialidades e WHERE e.des_especialidade = 'Obstetrícia' AND p.nr_sequencial NOT IN
```

```
(SELECT c.id_paciente FROM consultas c INNER JOIN medicos m ON  
c.id_medico=m.num_mec NATURAL JOIN especialidades e WHERE e.des_especialidade <>  
'Obstetrícia');
```

```
SELECT p.* FROM pacientes p  
INNER JOIN consultas c ON p.nr_sequencial=c.id_paciente  
INNER JOIN medicos m ON c.id_medico=m.num_mec  
NATURAL JOIN especialidades e  
WHERE e.des_especialidade = 'Obstetrícia' AND NOT EXISTS  
(SELECT c.id_paciente FROM consultas c INNER JOIN medicos m ON  
c.id_medico=m.num_mec NATURAL JOIN especialidades e WHERE e.des_especialidade <>  
'Obstetrícia' AND p.nr_sequencial=c.id_paciente);
```