

```

-- alinea a)
DELIMITER //
CREATE PROCEDURE `Atualiza_Precos_Especialidade_1` (IN ano INT, IN
especialidade varchar(100))
BEGIN
DECLARE preco_medio INT;

SET preco_medio = (select ROUND(AVG(a4.preco)) AS preco_medio
                    from (SELECT YEAR(data_hora) AS
ano,a1.preco,a3.designacao
                        FROM consulta a1,medico a2,
especialidade a3
                        where a1.id_medico=a2.id_medico
                        and
a2.id_especialidade=a3.id_especialidade
                        and YEAR(data_hora)=ano and
a3.designacao=especialidade) a4
                    group by a4.ano,a4.designacao);

update especialidade set preco=preco_medio where
designacao=especialidade;

END //
DELIMITER ;

CALL Atualiza_Precos_Especialidade_1(2015,'Oftalmologia');

DELIMITER //
CREATE PROCEDURE `Atualiza_Precos_Especialidade_2` (IN ano INT, IN
especialidade varchar(100))
BEGIN

UPDATE especialidade AS esp SET preco=(select ROUND(AVG(a4.preco)) AS
preco_medio
                    from (SELECT YEAR(data_hora) AS
ano,a1.preco,a3.designacao
                        FROM consulta a1,medico a2,
especialidade a3
                        where a1.id_medico=a2.id_medico
                        and
a2.id_especialidade=a3.id_especialidade
                        and YEAR(data_hora)=ano and
a3.designacao=especialidade) a4
                    group by a4.ano,a4.designacao) where
designacao=especialidade;
END //
DELIMITER ;

```

```

CALL Atualiza_Precos_Especialidade_2(2015,'Oftalmologia');

-- alinea b)
DELIMITER //
CREATE FUNCTION `Remove_medico_inativo` (id_medico INT) RETURNS
BOOLEAN
BEGIN

DECLARE num_consultas_medico INT DEFAULT 0;

SELECT COUNT(*) INTO num_consultas_medico FROM consulta WHERE
consulta.id_medico=id_medico;

IF num_consultas_medico = 0 THEN RETURN FALSE;
ELSE RETURN TRUE;
END IF;

END //
DELIMITER ;
/* EXECUTAR OPERACAO */

delete from medico where Remove_medico_inativo(id_medico)=0;
--OU
delete from medico where NOT Remove_medico_inativo(id_medico);

--alinea c)

-- (d) valor total faturado até ao momento por especialidade
SELECT SUM(c.preco) as valor_faturado FROM consulta c, medico m
where c.id_medico=m.id_medico
and m.id_especialidade=1
GROUP BY m.id_especialidade;

update especialidade e,
(SELECT SUM(c.preco) as valor_faturado,m.id_especialidade
FROM consulta c, medico m
WHERE c.id_medico=m.id_medico
GROUP BY m.id_especialidade) valor_faturado_per_especialidade
set e.valor_faturado=valor_faturado_per_especialidade.valor_faturado
where e.id_especialidade=valor_faturado_per_especialidade.id_especialidade;

-- (e) trigger after insert sobre a tabela consulta
DELIMITER $$
CREATE TRIGGER ATUALIZAR_VALOR_FATURADO_PER_ESPECIALIDADE
AFTER INSERT

```

```
ON CONSULTA
FOR EACH ROW
BEGIN
update especialidade e, medico m
set e.valor_faturado = e.valor_faturado + new.preco
where m.id_especialidade=e.id_especialidade
and m.id_medico=new.id_medico;
END $$
DELIMITER ;
```

```
-- Exemplo de insert de nova consulta
INSERT INTO CONSULTA (id_medico, id_paciente, data_hora, preco) VALUES
(123456, 123456, '2021-05-18 13:30', 200);
```

```
-- (f) Nova tabela consultas_per_especialidades
create table consultas_per_especialidades (
id_especialidade int primary key not null,
designacao varchar(200) not null,
num_consultas int,
valor_faturado int
);
```

```
-- (g) número e valor total faturado até ao momento por especialidade
SELECT count(c.preco) as num_consultas,SUM(c.preco) as
valor_faturado,m.id_especialidade, e.designacao
FROM consulta c, medico m, especialidade e
where +c.id_medico=+m.id_medico
and +m.id_especialidade=+e.id_especialidade
GROUP BY e.id_especialidade;
```

-- ou

```
SELECT count(c.preco) as num_consultas,SUM(c.preco) as
valor_faturado,m.id_especialidade, e.designacao
FROM consulta c
inner join medico m on m.id_medico=c.id_medico
inner join especialidade e on e.id_especialidade=m.id_especialidade
GROUP BY e.id_especialidade;
```

-- para adicionar a neurologia

```
INSERT INTO consultas_per_especialidades
(id_especialidade,designacao,num_consultas,valor_faturado)
SELECT e.id_especialidade, e.designacao, count(c.preco) as num_consultas,
CASE WHEN SUM(c.preco) IS NULL THEN 0 else SUM(c.preco) END as
valor_faturado
FROM consulta c
inner join medico m on m.id_medico=c.id_medico
right join especialidade e on e.id_especialidade=m.id_especialidade
GROUP BY e.id_especialidade;
-- confirmar o num de consultas por especialidade
```

```
select count(*) FROM consulta c, medico m
where c.id_medico=m.id_medico
and m.id_especialidade=2;
```

```
-- (h) trigger after insert sobre a tabela consulta sobre o num_consultas e
valor_faturado
```

```
DELIMITER $$
```

```
CREATE TRIGGER ATUALIZAR_CONSULTAS_PER_ESPECIALIDADE AFTER
INSERT
```

```
ON CONSULTA
```

```
FOR EACH ROW
```

```
BEGIN
```

```
update consultas_per_especialidades e, medico m
set e.valor_faturado = e.valor_faturado + new.preco,
    e.num_consultas=e.num_consultas+1
```

```
where m.id_especialidade=e.id_especialidade
and m.id_medico=new.id_medico;
```

```
END $$
```

```
DELIMITER ;
```

```
-- Exemplo de insert de nova consulta
```

```
INSERT INTO CONSULTA (id_medico, id_paciente, data_hora, preco) VALUES
(223456, 123456, '2021-05-18 14:30', 50);
```